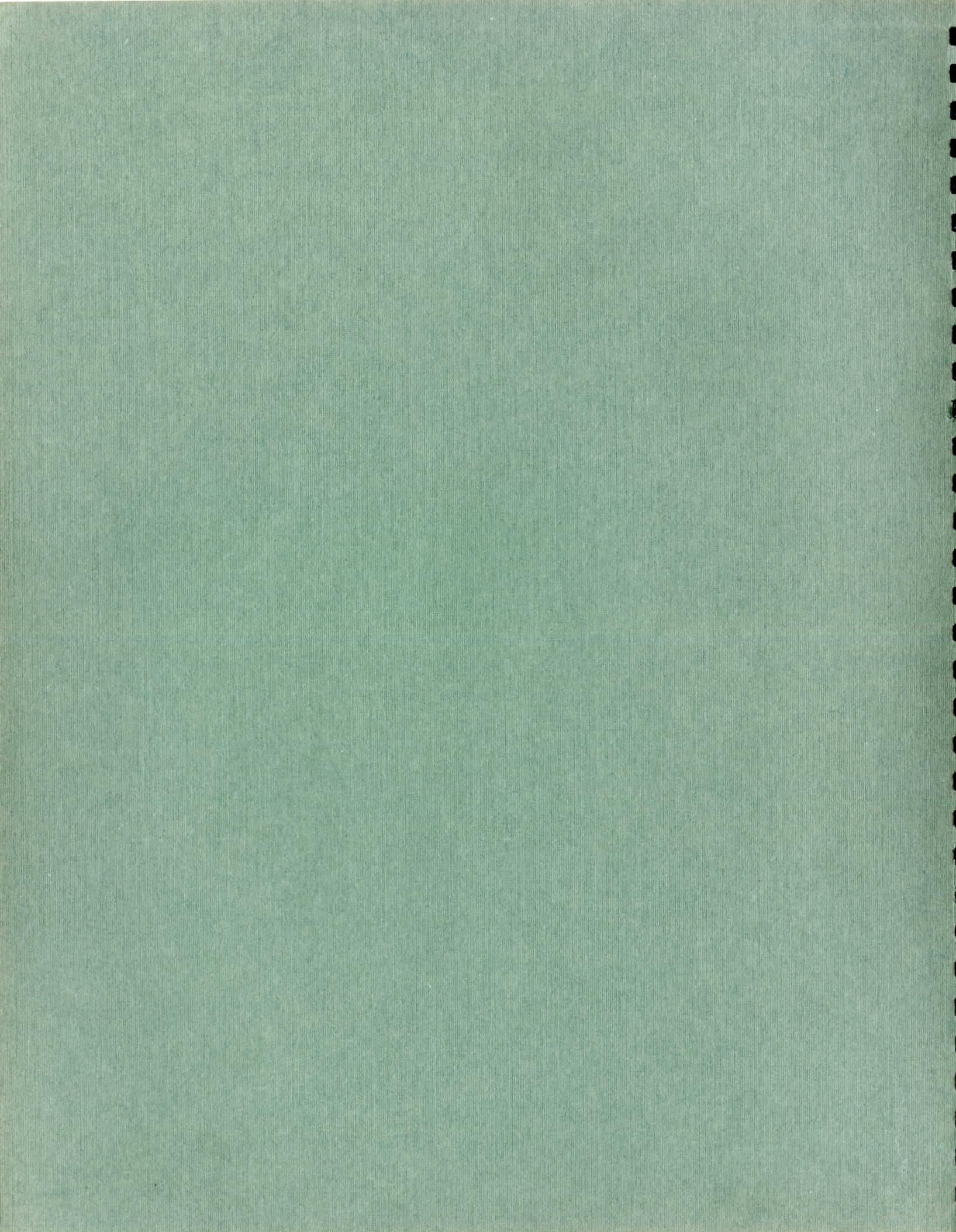


# The Smiffer<sup>TM</sup>

Token-Ring Network  
Portable Protocol Analyzer

## Operation and Reference Manual

Network  
General



# The Sniffer<sup>TM</sup>

Token-Ring Network  
Portable Protocol Analyzer

Operation and Reference  
Manual

Network  
General

## **DISCLAIMER OF WARRANTIES**

*The information in this document has been reviewed and is believed to be reliable; nevertheless, Network General Corporation makes no warranties, either expressed or implied, with respect to this manual or with respect to the software and hardware described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. The entire risk as to its quality and performance is with the buyer. The software herein is transferred "AS IS."*

*Network General Corporation reserves the right to make changes to any products described herein to improve their function or design.*

*In no event will Network General Corporation be liable for direct, indirect, incidental or consequential damages at law or in equity resulting from any defect in the software, even if Network General Corporation has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.*

*This document is copyrighted and all rights are reserved. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Network General Corporation.*

*The Sniffer* is a trademark of Network General Corporation.

*COMPAQ* is a trademark of COMPAQ Computer Corporation.

*IBM Token-Ring Network* is a trademark of IBM Corporation.

*©Copyright 1986 by Network General Corporation. All rights reserved.  
Present copyright law protects not only the actual text, but also the "look and feel"  
of the product screens, as upheld in the Atari and Broderbund cases.*

*Manual prepared by Paul Berry  
Appendices by Leonard J. Shustek  
December 1986*

# Contents

<b>Chapter 1. Overview .....</b>	<b>1</b>
<b>What the Sniffer Does .....</b>	<b>1</b>
The Sniffer Is Self-Contained .....	1
Menu-Driven Controls.....	1
Color Monitor .....	2
The Sniffer Is a Specialized Station on the Network .....	2
The Sniffer Copies Every Frame.....	3
Capture Filters .....	3
Real-Time Meters and Counters.....	4
<b>The Capture Buffer .....</b>	<b>4</b>
<b>The Trigger Detector Scans Incoming Frames .....</b>	<b>4</b>
A Trigger Event Stops Collection and Freezes the Buffer.....	5
Specifying the Trigger Pattern.....	6
Frames Surrounding the Trigger Frame .....	6
<b>Displaying the Frames in the Capture Buffer .....</b>	<b>6</b>
Saving the Capture Buffer for Later Analysis .....	6
Selecting the Form of Display .....	7
Windows in the Display .....	7
Two-Station Display .....	8
Dual Viewports.....	8
Saving and restoring setups.....	8
<b>The Frame Interpreters .....</b>	<b>9</b>
<b>Schematic View .....</b>	<b>9</b>
Key to Figure 1-1.....	11
<b>Chapter 2. The Sniffer at Work .....</b>	<b>13</b>
Live Examples.....	13
<b>Example 1: Slow Delivery of a Message .....</b>	<b>13</b>
Collecting Data .....	13
Checking MAC Frames.....	14
Monitor Frames.....	14
Names in the Display .....	15
Relative Time .....	15
Looking for SMB frames.....	16
Symbolic Station Names .....	16
Logical Link Control .....	17
Station-to-station Polling .....	18
NETBIOS Frames that Transmit the Message.....	18
Phases in Transmission of the Message .....	20

Search for Forwarded Address.....	20
Request for Name without Forwarding .....	21
The Message Text .....	22
Conclusions from the Inquiry .....	22
<b>Example 2:</b>	
<b>Inserting a New Station into the Ring .....</b>	<b>23</b>
Ring Purge .....	23
Duplicate Address Test.....	24
Change in Upstream Neighbor Address .....	24
Initialization Request .....	25
Active Monitor's Final Report of the Insertion.....	25
<b>Example 3:</b>	
<b>Batch File on a Server's Disk.....</b>	<b>26</b>
A Test File .....	26
Metering Data Flow .....	26
SMB Overview .....	27
Repeated Open and Close.....	28
Questions Arising from the SMB Display.....	28
Overall Communication Density.....	28
A Detailed Look at the Search for the Batch File .....	30
Reading the First Line of the Batch File.....	34
Play it Again, Sam .....	37
Questions Raised .....	39
<b>Chapter 3. Setting Up the Sniffer.....</b>	<b>41</b>
Unpacking.....	41
Hardware.....	41
Cables .....	42
Color Monitor Option .....	43
Software.....	44
<b>Starting the Sniffer .....</b>	<b>44</b>
Color, Resolution, and Brightness.....	45
<b>First Time Precautions .....</b>	<b>45</b>
<b>Organization of Software on the Hard Disk.....</b>	<b>46</b>
The Autoexec File.....	46
The Sniffer's Directory Conventions.....	47
Several Directories for Capture Files .....	47
Several Names Files.....	48
Updating the Current Names File.....	48
Loading or Saving Files from Another Directory .....	48
Backing Up the Software.....	49
Making a Bootable Sniffer Diskette.....	49

<b>The Sniffer's Main Menu .....</b>	<b>50</b>
A Tree-Structured Menu.....	51
A Movable Viewport.....	51
The Initial Menu.....	52
Preparing to Capture.....	52
Preparing to Display.....	52
To Conclude Work.....	52
<b>Conventions in the Sniffer Menus.....</b>	<b>53</b>
Function Keys .....	53
<b>Chapter 4. Capturing Frames .....</b>	<b>55</b>
Inserting the Sniffer into the Ring.....	55
Setting the Capture Filters .....	56
Station Address Filters .....	56
Protocols in the Capture Filter .....	59
Pattern Matching in the Capture Filter .....	60
<b>Setting the Trigger .....</b>	<b>62</b>
When to Stop Capture	
Positioning the Trigger Frame in the Capture Buffer.....	65
Marking the Trigger Frame .....	66
Stopping When the Buffer is Full .....	67
Continuous Capture .....	67
Totals .....	68
Traffic Counts .....	69
Pairwise Tabulation .....	69
Tabulation by Sender.....	70
Counting Frames, Kilobytes, or Percentage Utilization .....	71
Real-time Traffic Density Bar Graph .....	71
Unrecognized Addresses Noted During Capture .....	71
<b>Options During Capture .....</b>	<b>72</b>
Options During Pause.....	72
<b>Chapter 5. Displaying and Interpreting the Captured Data.....</b>	<b>75</b>
Deciding Which Set of Captured Data to Display.....	75
Loading a File of Previously-Saved Frames.....	76
Switching to Another Directory from within the List of Files .....	77
Setting a Path to a Different Directory.....	78
Creating a New Directory .....	78
Numbering of Frames in the Capture Buffer .....	78
Options Apply Equally to Screen and to Printer Displays .....	79
<b>Display Filters</b>	
Selecting Which Frames to Display.....	79
Timing: When You Can Set Filters and Displays.....	79
Procedure for Setting Display Filters .....	80

<b>Setting the Form of Display .....</b>	<b>82</b>
Hexadecimal view.....	82
Detail view .....	84
Address Recognized and Frame Copied Bits .....	85
Displaying Both Detail and Hexadecimal Listing.....	86
Summary View .....	87
Summary View in Two-Station Format .....	88
How the Sniffer Knows	
Which Stations to Show in Two-Station Format .....	88
Multiple Levels with the Summary View .....	89
Two Viewports Side-by-side .....	90
The Active Window .....	92
Highlighting Detail in the Hex Window .....	93
Scrolling Within a Window.....	93
Scrolling to Next Frame.....	93
Zooming for an Enlarged View of the Active Window .....	94
Selecting the Focus of Display .....	94
Moving Directly to Another Frame .....	95
Go to Frame Number .....	96
Jumping to a Frame that's Filtered Out.....	96
Searching for a Pattern .....	96
<b>How Time Is Displayed.....</b>	<b>97</b>
Absolute time .....	97
Delta time .....	97
Relative time .....	97
Network Utilization .....	98
<b>Managing Names Used in Displays and Filters .....</b>	<b>99</b>
The Sniffer Assigns a Name for Itself.....	100
Deducing What the Stations Call Themselves.....	100
Editing the Names Table .....	101
<b>Printing a Report on Frames in the Capture Buffer.....</b>	<b>102</b>
<b>Saving the Capture Buffer to a File .....</b>	<b>104</b>
Saving Your Current Setup .....	105
Using a Saved Setup File.....	105

## **Appendices**

<b>A. Format of Saved Data Files.....</b>	<b>107</b>
<b>B. File Name Conventions .....</b>	<b>111</b>
<b>C. Extending Sniffer Protocol Interpreters .....</b>	<b>113</b>
<b>D. Token-Ring Network Architecture.....</b>	<b>121</b>
<b>E. Glossary of Acronyms .....</b>	<b>127</b>
<b>F. Sniffer Specifications.....</b>	<b>133</b>
<b>G. References .....</b>	<b>135</b>
<b>H. Troubleshooting Checklist.....</b>	<b>137</b>
<b>Index.....</b>	<b>143</b>



# List of Figures

1-1	Schematic representation of the Sniffer's functions.....	10
2-1	Summary view of MAC frames.....	14
2-2	MAC frames displayed with time relative to frame 18.....	15
2-3	Display with everything but SMB frames filtered out.....	16
2-4	Logical link control frames between the two stations, but before one sent a message to the other.....	17
2-5	LLC frames containing NETBIOS frames that start transmission of the message.....	18
2-6	Summary and Detail views of logical link control frames at the start of the message.....	19
2-7	Summary view of NETBIOS frames regarding transmission of the message.....	19
2-8	Summary and Detail views of frames in which station Mary seeks to identify station Tom in order to send a message.....	20
2-9	Transmission of the message. ....	21
2-10	Using two viewports to compare the frames in which the message is transmitted.....	22
2-11	MAC frames surrounding insertion of a new station into the ring. ....	24
2-12	Detail view of frame showing report of upstream neighbor's address. ....	25
2-13	Meters and counters recordings traffic between two stations during test. ....	26
2-14	Printer output, execution of a batch file, SMB frames.....	27
2-15	LLC and SMB frames, with mark set at the start of the sequence (frame 3).....	29
2-16	Summary view of LLC and SMB frames (showing all levels), with time relative to the start of the batch file sequence. ....	29
2-17	Percentage of network bandwidth utilized by the selected frames during a 100-millisecond window around each frame. ....	30
2-18	Request to search for file TEST on Server machine. ....	31

2-19	Server's reply to the search request.....	31
2-20	User machine says it was unable to receive. ....	32
2-21	The User machine says it is now ready to receive the balance of the transmission.....	32
2-22	Server repeats the name of the file it has found. ....	33
2-23	The User machine requests continuation of the search for the file TEST. ....	33
2-24	Server says it can no find no more entries for TEST.???.	34
2-25	The User machine's request to open the batch file.....	35
2-26	Display of frame 29, zoomed so the Detail view has the entire screen, showing the Server's reply to the request to open file TEST.BAT. ....	35
2-27	The User machine asks for the first 512 bytes of the file TEST.....	36
2-28	Server transmits all 52 bytes of file TEST.BAT.....	36
2-29	The User machine is unable to accept the transmission.....	37
2-30	The User machine is ready for the rest of the file TEST.BAT.....	37
2-31	The Server repeats transmission of the file TEST.BAT.....	38
3-1	Connections to the Sniffer's adapter cards.....	43
3-2	The first panel of the Sniffer's Main Menu.....	50
4-1	Default settings of Capture Filters for station address.....	56
4-2	Menu to select a station for a station address filter. ....	57
4-3	Window for inserting a new name and station address. ....	58
4-4	Menu to select SAPs for the Capture Filter. ....	59
4-5	Specifying pattern match for the Capture Filter. ....	60
4-6	Inserting a pattern and offset for the Capture Filter. ....	61
4-7	Default settings of the Trigger. ....	62
4-8	Window to supply offset while specifying a trigger pattern. ....	63
4-9	Window in which to supply Trigger pattern.....	64
4-10	Specifying whether the trigger pattern's offset is frame- or data-relative.....	65

4-11	Selecting the rule for stopping capture. ....	66
4-12	Capture option in the main menu. ....	68
4-13	Pairwise tabulation during capture, by sending station and addressee. ....	69
4-14	Individual tabulation by sending station during capture. ....	70
5-1	The main menu, showing the Display option and its principal branches. ....	75
5-2	Main menu, showing choices you select to load the Capture Buffer with data from a file. ....	76
5-3	List of saved files that can be loaded to the Capture Buffer. ....	77
5-4	Menu to establish display filters. ....	80
5-5	Display filters menu, showing list of protocol levels. ....	81
5-6	Hexadecimal view of a frame. ....	82
5-7	Menu to select the translation of hex characters. ....	83
5-8	Part of the Detail view of the frame that was shown in hexadecimal in Figure 5-6. ....	84
5-9	Scrolling reveals other levels of detail in the same frame. ....	85
5-10	Detail and hexadecimal views of the same frame, shown in two windows. ....	86
5-11	Summary view, showing frame 35 in the context of neighboring frames. ....	87
5-12	Two-station form of the summary view. ....	88
5-13	Summary display with the highest-level-only restriction removed (here shown in two-station format). ....	89
5-14	Menu to select two independent side-by-side viewports, superimposed summary display. ....	90
5-15	Display with two viewports, each containing a summary window and a detail window. ....	91
5-16	Two viewports, each with three windows. ....	92
5-17	Superimposed menu showing options for moving around in the Capture Buffer. ....	95
5-18	Superimposed screen on which to write the number of the frame to which you want to go. ....	96

5-19	Specifying a pattern to jump to.....	97
5-20	Menu to select the form of time display, superimposed on the summary window, showing additional options for average network utilization.....	98
5-21	Menu options for managing names used in Sniffer displays.....	99
5-22	Display of the names table.....	101
5-23	Window to provide a new symbolic name for a station.....	102
5-24	Menu to select printing of a report on frames in the Capture Buffer.....	103
5-25	Menu for saving data files or setup files. ....	104

# Chapter 1. Overview: What the Sniffer Does

*You can observe a lot by watching.*  
--Yogi Berra.

The Sniffer collects, analyzes and interprets data circulating in a token-ring network.

It permits detailed analysis of network transmissions at all levels, from the data link level on which everything else rests, up to the session level used by network applications. With its detailed records of exactly what transpires during network transactions, it is a powerful tool for trouble-shooting and tuning a network, and for testing and refining high-performance network software.

## **The Sniffer Is Self-Contained**

The Sniffer is a fully portable computer and is completely self contained. It comes with its own token-ring adapter already installed, its own hard disk, and its own operating system and software ready to run.

To start using the Sniffer, you need only plug its power cord to a suitable outlet and its token-ring cable to the network.

The only customizing you'll probably find desirable is to augment the Sniffer's definition file of station names. The Sniffer can then display both the hardware addresses it observes and the names by which you refer to the various machines. (You can do that as you go along; see *Managing Names* in Chapter 5.)

## **Menu-Driven Controls**

An autoexec batch file already installed on the hard disk starts the Sniffer software as soon as you turn the machine on.

You operate the Sniffer from a menu screen. You move the cursor to the choices you want, select options by pressing the space bar while they're highlighted, and press *Enter* or one of the function keys to start an action. Whenever a function key is operative, it's highlighted and labeled in the screen display.

There is no command language, and there are no commands to learn. About the only information you'll supply by typing is the name for a file you wish to save.

When you exit from the Sniffer's software, it returns you to its operating system, COMPAQ DOS 3.10. The Sniffer is then a standard AT-class personal computer operating under DOS. A DOS manual is included with the Sniffer.

## **Color Monitor**

The Sniffer's built-in monitor is high-resolution 8-level grey-scale monochrome, with a green phosphor. You can also connect your own color monitor. The Sniffer provides a DB-9 jack for an RGBI monitor that supports the IBM color graphics monitor, and an RCA jack for composite video. You have only to plug in your equipment and report that it's there when the Sniffer program asks.

## **The Sniffer Is a Specialized Station on the Network**

Like most of the other network devices, the Sniffer is an independent computer with its own software and hardware, and its own token-ring adapter. It does not need (and does not include) a copy of the network management software used by ordinary stations on the network.

The Sniffer comes with its own DB9-to-token-ring cable. You connect its token-ring adapter to the network in the same way that you connect any other station: by simply plugging it to any available access unit on the network.

As far as the other stations on the network are concerned, the Sniffer is a passive member. Like any ordinary station on the network, it receives transmissions from the neighboring upstream station, and immediately relays each of them to the neighboring downstream station. On every network, one of the stations must play the role of active monitor (generating clock signals, regenerating a lost token, and so on). Every station must be ready to take on that role when no other station is doing so, and the Sniffer is no exception. But the Sniffer doesn't otherwise originate traffic addressed to other stations, and never acts as receiver for messages they send. Although the Sniffer is visible to other machines (for example, when they poll all stations that are connected), it takes no other part in network business.

## **The Sniffer Copies Every Frame**

Like every other station on the network, the Sniffer retransmits every frame it receives, relaying it to the station downstream from it. However, as it repeats each incoming frame, the Sniffer's adapter card makes a temporary duplicate. It passes each of the duplicate frames to the Sniffer's on-board processor for review. All the rest of this manual concerns the duplicate frames that the Sniffer retains. The Sniffer sifts these duplicate frames, and records some of them for analysis and later interpretation.

## **Capture Filters**

The number of frame passing through a token-ring adapter is potentially so large that it's essential to select only a subset. The Sniffer applies a filter to each newly-arrived frame, and discards the frames that do not meet its test. Capture Filters are of three types:

- Selection by station address: include frames sent from or received by a particular station or pair of stations.
- Selection by protocol: include frames containing any of the protocols you specify.
- Selection by pattern: include frames containing a specified pattern of data at a particular position in the frame.

(For example, a typical filter might admit only messages to or from a particular user and a server with which the user is experiencing a problem, and only those frames involving NETBIOS or SMB protocols).

Setting an appropriate filter is your first step in collecting data. Often, the majority of arriving frames are discarded at once. The frames that your filters admit then pass to a buffer area, from which you may display them, send them to storage, or discard them.

## **Real-Time Meters and Counters**

While the Sniffer is collecting data, it measures the rate at which frames are arriving, and gives you a real-time graphic display of *meters* (which show the data-rate), and *counters* (which show a running total of the numbers of frames transmitted).

You can display the traffic density as kilobytes per second or as frames per second. For either, you can elect to show them as absolute values or as percentages of the network's available bandwidth, and on a linear or on a logarithmic scale.

Counters can tabulate frames by destination or by source, or cross-tabulate them by station pairs. The display is expanded in real time. As the Sniffer notices traffic involving stations it hasn't seen before, it makes room in the display to include them.

## **The Capture Buffer**

After they've been counted, frames that the filter accepts pass to the Capture Buffer. (On the way, they're examined by the Trigger Detector, described in a moment.) The Capture Buffer has room for a moderate number of frames (hundreds of medium-sized frames, or thousands of minimal-sized ones). Frames accumulate in the Buffer in the order they are received.

When the Capture Buffer becomes full, the Sniffer discards older frames to make way for new arrivals. If you do nothing to retain the frames in the Capture Buffer, the Sniffer automatically discards them; in that case, the frames that remain in the Buffer are the ones most recently received.

## **The Trigger Detector Scans Incoming Frames**

The Trigger Detector scans the stream of incoming frames. It's located after the Capture Filter, so that it looks only at frames that have passed through the filter but haven't yet reached the Capture Buffer (see Figure 1-1).

The Trigger Detector looks for a frame containing a particular pattern that you've described. When it finds such a frame, it signals a trigger event. The trigger event freezes the Capture Buffer so you can examine the trigger frame and the frames that precede or follow it.

## A Trigger Event Stops Collection and Freezes the Buffer

When the Trigger Detector signals a trigger event, capture ceases, either immediately or with enough delay to collect some of the following frames. Once capture has been halted, you can:

- Copy the contents of the Capture Buffer to a file for later analysis or display.
- Browse through various displays of the frames in the Capture Buffer.
- Impose a display filter to select which frames are displayed.
- Select one or more views (ways of displaying a frame).
- Print the contents of the buffer, according to the filters and views you've specified.

A trigger event halts the processing of incoming data. It causes the Sniffer to cease capturing frames until you say you're again ready to receive them.

## **Specifying the Trigger Pattern**

A trigger pattern is a set of characters at a particular position in a frame. You can describe the position either absolutely, or relative to the start of the frame's data field. You can make the test match either the presence or the absence of the pattern.

For example, if you're examining complaints of intermittent problems with access to a particular file server, you set up a collection filter that accepts only frames to or from that station, and a trigger that signals when it spots an error return code.

The frame that matches the trigger pattern is called the *trigger frame*. When it appears in your display of the Capture Buffer, the trigger frame is identified by a letter T beside it. One of the actions during display is "Jump to trigger frame."

## **Frames Surrounding the Trigger Frame**

When you set up a trigger pattern, you also indicate where in the Capture Buffer you want the trigger frame to appear. That determines whether the Buffer contains frames that preceded the trigger frame, frames that followed it, or some on either side.

## **Displaying the Frames in the Capture Buffer**

You have many options for displaying the contents of the Capture Buffer, either at the Sniffer's screen or to a printer. (You can direct printer output either to a locally-attached printer, or to a file on one of the Sniffer's disk drives.)

You can set up a display filter so that frames which don't interest you are omitted from the display (even though they remain in the Capture Buffer). The mechanism for filtering frames from the Capture Buffer is the same as the mechanism for filtering frames during capture.

## **Saving the Capture Buffer for Later Analysis**

From the keyboard, you can select a command that saves the contents of the Capture Buffer to a file. You can save the entire Capture Buffer, or just the frames that are selected by your current Display Filter.

All displays describe the data in the Capture Buffer. You can display data that has arrived and is still in the Buffer, or you can load the Capture Buffer with data you earlier saved to a file.

## Selecting the Form of Display

The display may contain any or all of the following three reports:

- **Hexadecimal view.** The entire frame is listed. Character data are displayed according to ASCII or EBCDIC conventions, as appropriate.
- **Detail view.** Each frame is decoded to show the type of frame and the values of its various fields. If you provide a file of symbolic names for station addresses, the *detail* view augments the station names with the symbolic names provided in your file of definitions.

For high-level frames, the interpretation may take several levels. The “higher” level interpretation of a frame is more deeply nested within it. The various interpretations are shown with the “higher” protocol levels (i.e. the ones that are more “deeply” embedded) after the lower ones.

- **Summary view.** This condensed form abbreviates and truncates some of the information from the hexadecimal view and some of the information from the *detail* view. It forces each level of interpretation to fit on a single line. The display contains one line for each level of protocol in the frame. You can elect to show only the highest level; in that case, the *summary* view has one line per frame.

The *hexadecimal* view and the *detail* view show data for just one frame. The *summary* view shows not only the frame you’re now examining, but a few on either side of it as well, to give context.

## Windows in the Display

Each view you elect appears in a window. The screen is divided into one, two or three equal-sized windows, one above the other, according to the number of views you request.

The window that contains the cursor bar is the active window. The Tab key moves the highlight from one window to the next, activating the window where it arrives. When a frame’s display won’t fit within its window, you can scroll the active window to see the information you want. You can also zoom in to the active window, temporarily, giving it the entire screen until you zoom out and restore the other windows.

## **Two-Station Display**

Frequently, analysis concerns the flow of commands back and forth between a pair of stations. In that situation, it is often helpful to elect *two-station display*. Frames from one station are shown on one side of the screen or paper, frames from the other on the other side. (Frames that are not part of the two-way interaction are also shown, but in the default format.)

## **Dual Viewports**

Sometimes it's important to compare a frame from one part of the Capture Buffer with a frame that arrived earlier or later. You can do that by electing dual viewports. The screen is split into left and right halves. In each window, you can scroll the two sides independently, permitting you to concentrate on one frame on the left and another frame on the right.

## **Saving and restoring setups**

Because there's a rich choice of options concerning what to display and how to display it, the Sniffer lets you save a record of the way you are filtering and displaying frames, so that you can readily restore the setup at a subsequent work session.

## The Frame Interpreters

The Sniffer doesn't just capture and store frames from the network. It also interprets them. When you select the *detail* view, for each frame you get a set of interpretations, one interpretation for each level of protocol that the frame contains. The interpreter labels and decodes the standard fields in each frame, making it easy to see the message conveyed.

When you select both a *detail* view and a *summary* view, the Sniffer automatically scrolls the *detail* view so that the interpretation shown there matches the level you've highlighted in the *summary* view.

If your network transmits frames whose protocol is unknown to the Sniffer's interpreters, it's possible to augment them with a custom interpreter of your own. To write one, you'll need detailed familiarity with the protocol, with the token ring's DLC and LLC conventions (data link control and logical link control), and the C programming language. Specifications for such an interpreter are provided in Appendix C.

## Schematic View

Figure 1-1 conceptualizes the Sniffer's various functions. It's more a cartoon than a formal diagram, but it correctly conveys the flow of data in the Sniffer.

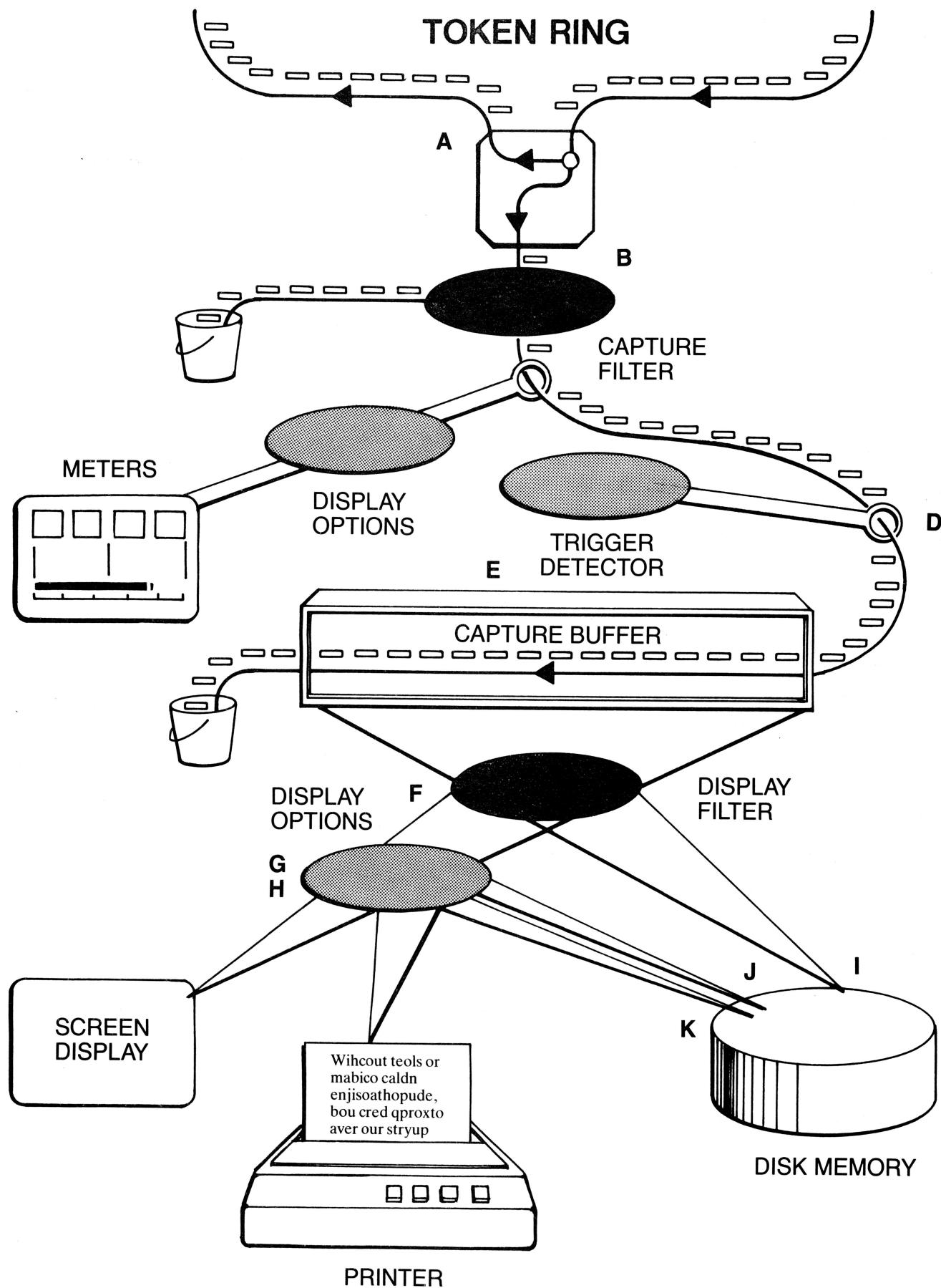


Figure 1-1: Schematic representation of the Sniffer's functions

## Key to Figure 1-1

- A. At the top, frames circulating on the ring are flowing in a clockwise direction. The Sniffer's specially-modified network adapter card serves both to relay frames to the next downstream station and to retain a copy for the Sniffer's analysis.
- B. The Capture Filter immediately discards frames that don't meet its address and protocol filters.
- C. During capture, the meters and counters provide real-time display of activity of the captured frames.
- D. The trigger scans the accepted frames for patterns for which it has been alerted.
- E. The capture buffer holds frames captured from the network, or loaded from a file. Unless capture is halted manually or by a trigger, as the buffer is filled by arriving frames, the earlier frames are discarded to make room for the new ones.
- F. Display filters select the frames visible on the screen or printer display.
- G. Display options select the number and type of views displayed.
- H. The frame interpreters decode fields in the frame being viewed.
- I. The set of frames in the capture buffer can be saved to a file (with or without winnowing by the display filter).
- J. Set-ups (display filters, view options, etc.) may also be saved to a file.
- K. A stored file of definitions helps interpret station names in the display.



# Chapter 2. The Sniffer at Work

To illustrate how the Sniffer works and show the kinds of things you can do with it, this chapter presents three worked examples. It describes the questions you might ask and the displays you'd see as you use the Sniffer to investigate operations on a network. The examples that follow show some of the reports in considerable detail, but don't dwell on the steps that generate them; discussions of the Sniffer's options and controls are in Chapters 4 and 5.

## Live Examples

Some of the examples reveal design errors, or at least less-than-optimal practices, in the software managing the transmissions. That, after all, is what the Sniffer is for: to isolate and describe problems. As you study these examples, you may be tempted to think we made them up, or deliberately wrote programs with errors so that we could show how to spot them. We didn't do that. The examples are not fictitious. They are taken from actual records of network sessions using unmodified versions of standard released software from major vendors.

### Example 1: Slow Delivery of a Message

Suppose your client is using an available software package that includes a facility for electronic messaging. Each user on the network is assigned a short name (for example, Tom, Mary, Harry, etc.). Any station can send another station a message containing a page or so of text. When the addressee's machine receives the message, it emits a beep; the recipient can display the text at once or file it for examination later. But users notice that it takes several seconds for each such message to arrive. On a network whose transmission rate is 4,000,000 bits per second, the delay seems surprising. The delay seems unchanged even when the network is essentially idle. What's the cause?

## Collecting Data

At a time when the network isn't much used, you connect the Sniffer to a point on the ring in sight of the machines used by Mary and Tom. You ask Mary to prepare a message to Tom, but not to send it yet. You ask Tom to sing out as soon as he sees the message. Then you do the following:

- Start collecting data at the Sniffer (without a collection filter).
- Tell Mary to send her message.
- Stop collecting data as soon as Tom reports that the message has arrived.

- Save the Capture Buffer to a file. (This isn't essential, but prudent if you may want to look at the data again later.)

It isn't necessary to confine this test to a quiet time on the ring, but that means you don't need a filter to weed out other traffic and thus simplifies this example.

## Checking MAC Frames

To verify that communication seems normal before the message is sent, you start with a *summary* view, filtering to show only the MAC frames. Figure 2-1 shows the first screenful.

SUMMARY	Delta t	DST	SRC	
4	4.799	Broadcast	+Tom	MAC Active Monitor Present
7	0.021	Broadcast	+This Sniffer	MAC Standby Monitor Present
8	0.019	Broadcast	+Mary	MAC Standby Monitor Present
11	6.957	Broadcast	+Tom	MAC Active Monitor Present
14	0.019	Broadcast	+This Sniffer	MAC Standby Monitor Present
15	0.011	Broadcast	+Mary	MAC Standby Monitor Present
18	6.967	Broadcast	+Tom	MAC Active Monitor Present
21	0.018	Broadcast	+This Sniffer	MAC Standby Monitor Present
22	0.012	Broadcast	+Mary	MAC Standby Monitor Present
25	6.967	Broadcast	+Tom	MAC Active Monitor Present
28	0.016	Broadcast	+This Sniffer	MAC Standby Monitor Present
29	0.014	Broadcast	+Mary	MAC Standby Monitor Present
32	6.967	Broadcast	+Tom	MAC Active Monitor Present
35	0.015	Broadcast	+This Sniffer	MAC Standby Monitor Present
36	0.015	Broadcast	+Mary	MAC Standby Monitor Present
39	6.982	Broadcast	+Tom	MAC Active Monitor Present
42	0.013	Broadcast	+This Sniffer	MAC Standby Monitor Present
43	0.017	Broadcast	+Mary	MAC Standby Monitor Present
54	6.967	Broadcast	+Tom	MAC Active Monitor Present
55	0.011	Broadcast	+This Sniffer	MAC Standby Monitor Present

1 Help    2 Set mark    5 Menus    6 Disp/ options    7 Prev frame    8 Next frame    10 New capture

Figure 2-1: Summary view of MAC frames.

## Monitor Frames

What you see in Figure 2-1 is the regular "heartbeat" of a token ring network. At regular intervals, one station broadcasts that it is the active monitor, and each of the others responds that it is a standby monitor. In this case, the machine called Tom is serving as active monitor. You can see that machines here called Tom, Mary, and the Sniffer are the only stations on the network.

## Names in the Display

At the level of MAC frames, stations address each other solely by the unique hardware IDs supplied in each network adapter. These names consist of 6 bytes (shown as 12 hex digits in the display). To make the listings easier to read, the Sniffer's display routines insert symbolic names such as "Tom," "Mary" or "Sniffer." The Sniffer automatically translates the hardware station address according to the information in a dictionary file. You may edit the file to supply convenient names (see Chapter 3); that's how we supplied the names *Tom*, *Mary*, and *This Sniffer*.

## Relative Time

By switching the display to Relative Time and marking one of the "active monitor" announcements so that it becomes Time 0, the regular rhythm of the monitor announcements becomes more obvious (Figure 2-2). It's evident that the active monitor sends its signal at an interval just a little less than 7 seconds, and the other stations respond (each in sequence in the direction downstream from the monitor) about a millisecond later.

SUMMARY	Rel time	DST	SRC	
	-6.998	Broadcast	+Tom	MAC Active Monitor Present
	-6.978	Broadcast	+This Sniffer	MAC Standby Monitor Present
	-6.967	Broadcast	+Mary	MAC Standby Monitor Present
M 18	0.000	Broadcast	+Tom	MAC Active Monitor Present
	0.018	Broadcast	+This Sniffer	MAC Standby Monitor Present
	0.030	Broadcast	+Mary	MAC Standby Monitor Present
	6.998	Broadcast	+Tom	MAC Active Monitor Present
	7.014	Broadcast	+This Sniffer	MAC Standby Monitor Present
	7.029	Broadcast	+Mary	MAC Standby Monitor Present
	13.996	Broadcast	+Tom	MAC Active Monitor Present
	14.011	Broadcast	+This Sniffer	MAC Standby Monitor Present
	14.027	Broadcast	+Mary	MAC Standby Monitor Present
	21.009	Broadcast	+Tom	MAC Active Monitor Present
	21.022	Broadcast	+This Sniffer	MAC Standby Monitor Present
	21.040	Broadcast	+Mary	MAC Standby Monitor Present
	28.007	Broadcast	+Tom	MAC Active Monitor Present
	28.019	Broadcast	+This Sniffer	MAC Standby Monitor Present
	28.038	Broadcast	+Mary	MAC Standby Monitor Present
	35.005	Broadcast	+Tom	MAC Active Monitor Present
	35.026	Broadcast	+This Sniffer	MAC Standby Monitor Present

1 Help    2 Set mark    5 Menus    6 Display options    7 Prev frame    8 Next frame    10 New capture

Figure 2-2: MAC frames displayed with time relative to frame 18.

The MAC frames shown here contain no error reports to suggest a hardware malfunction at either station, so it seems reasonable to focus on the frames devoted to sending and receiving the message itself.

## Looking for SMB frames

Sending a message is handled by frames containing data at the SMB (“System Message Block”) level. Is the message visible if you filter out everything but frames containing SMB traffic? Figure 2-3 show such a display.

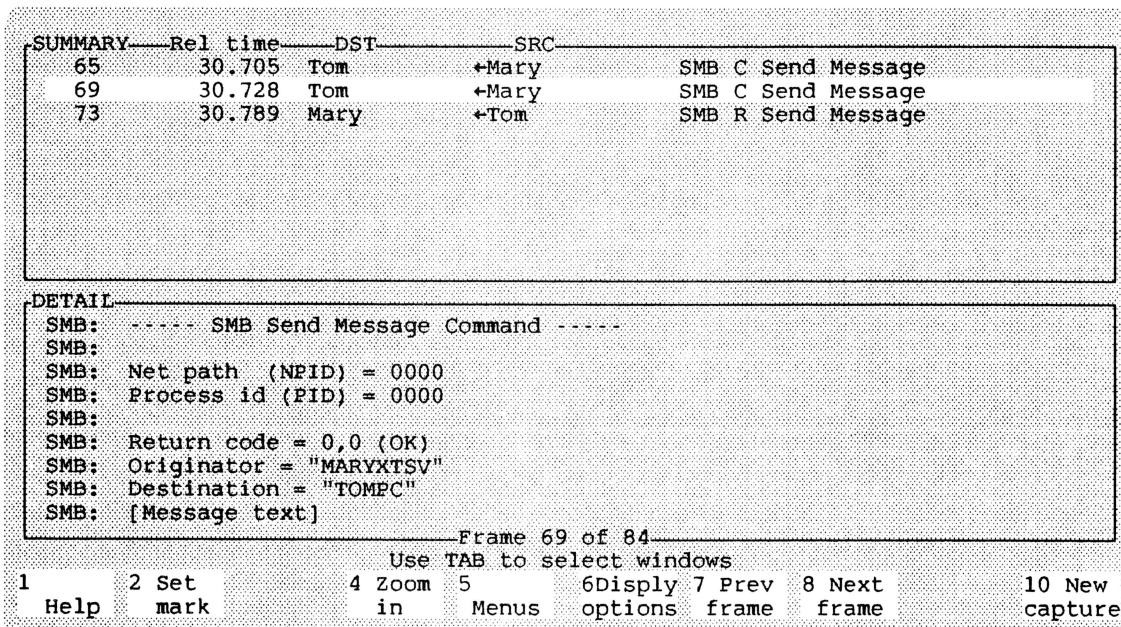


Figure 2-3: Display with everything but SMB frames filtered out.

The SMB display does indeed show the message. In fact, it shows two message transmissions (“C”). .02 seconds apart, followed by a single acknowledgment (“R”). .06 seconds after that.

This display doesn’t give much clue to the possible causes of sluggish response to the message command, except for the puzzling fact that there are two *send* commands from Mary to Tom.

## Symbolic Station Names

On the network there are frequently two kinds of names for machines. There are hardware addresses, fixed by the unique hardware of each machine’s network adapter card. There are also symbolic names that the network software may permit machines to adopt.

The Sniffer’s protocol interpreters supply their own symbolic names during display. To make it easier to read the hardware addresses, at startup the Sniffer consults a file of definitions. The file gives more readable synonyms for the various hardware addresses. For example, the file supplied with each Sniffer provides the synonym “This Sniffer” for that particular machine’s hardware address.

Because the symbolic names supplied by the protocol interpreter are taken from a text file that you can edit at will, they're really independent of the "log-on" names that the network software may permit machines to adopt. (Indeed, the network may permit a machine to respond to several different names.) In Figure 2-3, you can see that the definition file used for this example supplied the names "Tom" and "Mary" for the two machines. However, the names that appear in the network SMB frames are TOMPC and MARYXTSV, presumably because those were the network names those machines had adopted during that work session.

The dictionary file is arbitrary. You can tell the Sniffer to display whatever characters you find convenient as its interpretation of the hardware addresses. Naturally, it's easier to understand the display when the names you put in the dictionary correspond to any names that appear in the transmissions.

## Logical Link Control

Since the MAC frames don't indicate a hardware problem, and the SMB frames don't account for the delay, you experiment next with frames at the logical link level. For this phase of the inquiry, you request display in Two-Station format. That makes it easy to examine a dialogue between two stations. It shows one station's frames on the left and the other station's on the right. At the same time, you adjust the filters to show only LLC frames. (Figure 2-4).

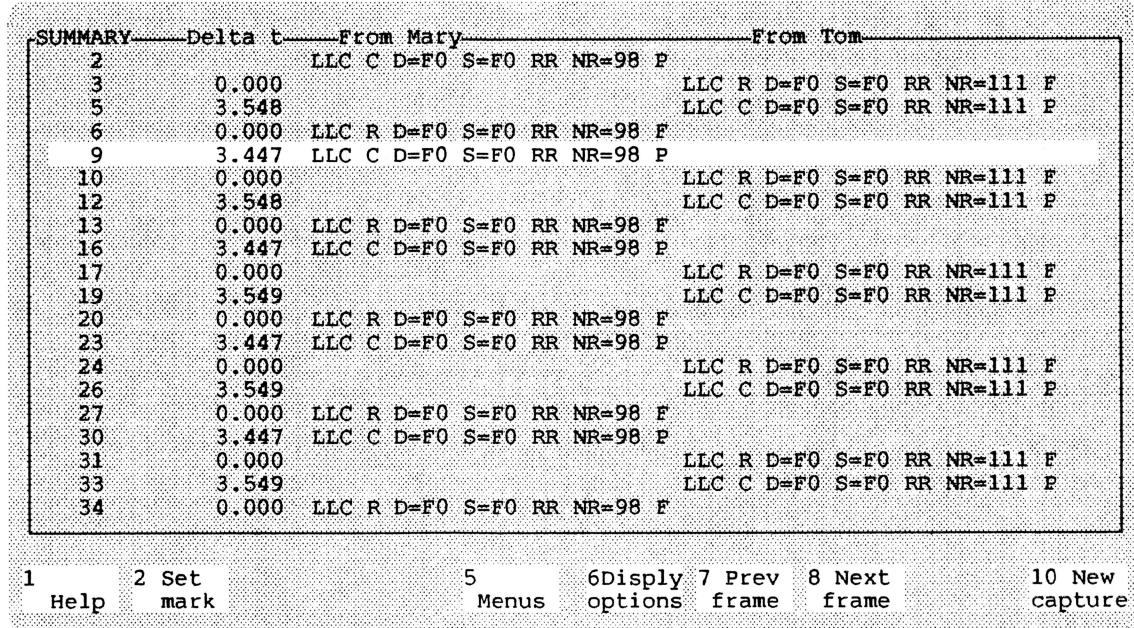


Figure 2-4: Logical link control frames between the two station, but before one sent a message to the other.

## Station-to-station Polling

The display shown in Figure 2-4 illustrates another “heartbeat” of the network. Station “Mary” has established a link with station “Tom.” (The link arose because Mary is acting as a server machine, and Tom is currently established as a user of a directory there, even though not at the moment making use of that directory.)

You can see that every 7 seconds one machine polls the other (LLC record ending in P) and the other immediately responds (LLC record ending in F). Both machines poll, but their polling is staggered, so that one or the other occurs about every 3.5 seconds. The frames are all RR frames, which indicate readiness for data transmission but don’t themselves contain any data. This too indicates a normal state of affairs, and doesn’t account for the message delay.

## NETBIOS Frames that Transmit the Message

Scrolling forward in the file, you come to some LLC frames which contain NETBIOS commands. Here’s the start of the effort to transmit a message from Mary to Tom (Figure 2-5).

SUMMARY	Rel time	From Mary	From Tom
40	38.555		LLC C D=F0 S=F0 RR NR=111 P
41	38.556	LLC R D=F0 S=F0 RR NR=98 F	
44	40.621	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
45	41.099	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
46	41.599	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
47	42.004	LLC C D=F0 S=F0 RR NR=98 P	LLC R D=F0 S=F0 RR NR=111 F
48	42.004		LLC C D=F0 S=F0 UI
49	42.099	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
50	42.599	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
51	43.098	LLC C D=F0 S=F0 I NR=98 NS=111	LLC R D=F0 S=F0 RR NR=112
52	43.101		LLC C D=F0 S=F0 UI
53	43.198	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
57	46.551		LLC C D=F0 S=F0 RR NR=112 P
58	46.552	LLC R D=F0 S=F0 RR NR=98 F	
59	48.203	NETBIOS ←Mary	LLC C D=F0 S=F0 UI
60	48.214		LLC C D=F0 S=F0 UI
61	48.224	LLC C D=F0 S=F0 I NR=98 NS=112	LLC R D=F0 S=F0 RR NR=113
62	48.227		LLC C D=F0 S=F0 I NR=113 NS=9
63	48.234		
64	48.238	LLC R D=F0 S=F0 RR NR=99	

1 Help    2 Set mark    5 Menus    6 Dispaly options    7 Prev frame    8 Next frame    10 New capture

Figure 2-5: LLC frames containing NETBIOS frames that start transmission of the message.

By opening a *detail* view in addition to the *summary* view, you can see that the first of the NETBIOS messages is a name inquiry, seeking the addressee of the message (Figure 2-6). To see a slightly longer summary of each command, you temporarily turn off the two-station format; to see the entire interpretation of each command, you have to scroll in the *detail* window, or zoom to devote the entire screen to the *detail* view.)

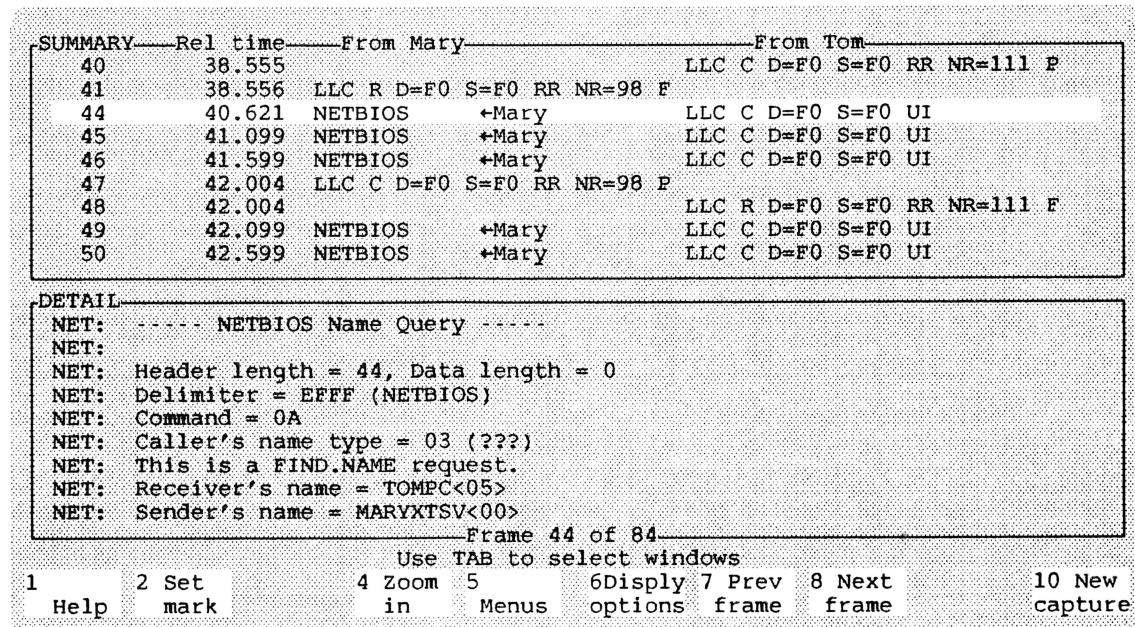


Figure 2-6: Summary and Detail views of logical link control frames at the start of the message.

The handling of the message itself is done in the succession of NETBIOS frames contained within some of these LLC frames. By switching the filter to NETBIOS rather than LLC, you further limit the display. Since frame 44 seems to be the start of the sequence, press F2 to mark it, so that relative time is calculated with respect to it as time zero. By pressing F4, you can zoom so that the *summary* view has the whole window (Figure 2-7).

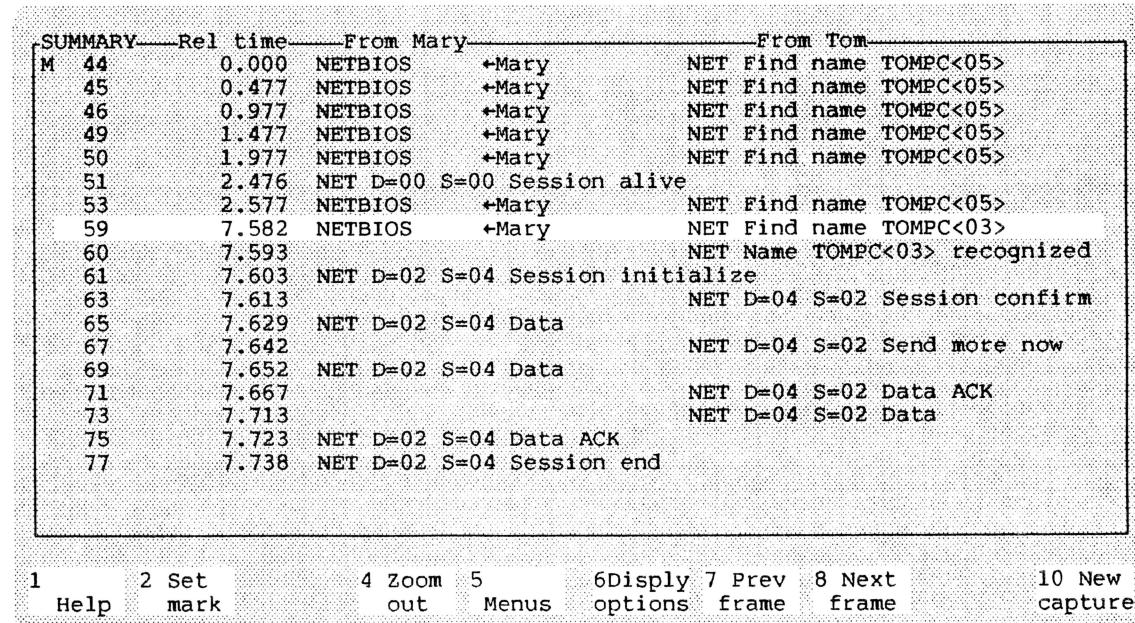


Figure 2-7: Summary view of NETBIOS frames regarding transmission of the message.

## Phases in Transmission of the Message

Events regarding the transmission seem to be in three phases. Frames 44 to 60 are searches for a name. Frames 61 and 63 initialize the session in which the message is transferred, and frame 77 concludes the session. Frames 65 to 75 deal with the actual transmission of the message.

In Figure 2-8 we take a closer look at the first phase, showing the *summary* and the *detail* view for the frames which attempt to locate the recipient of the message .

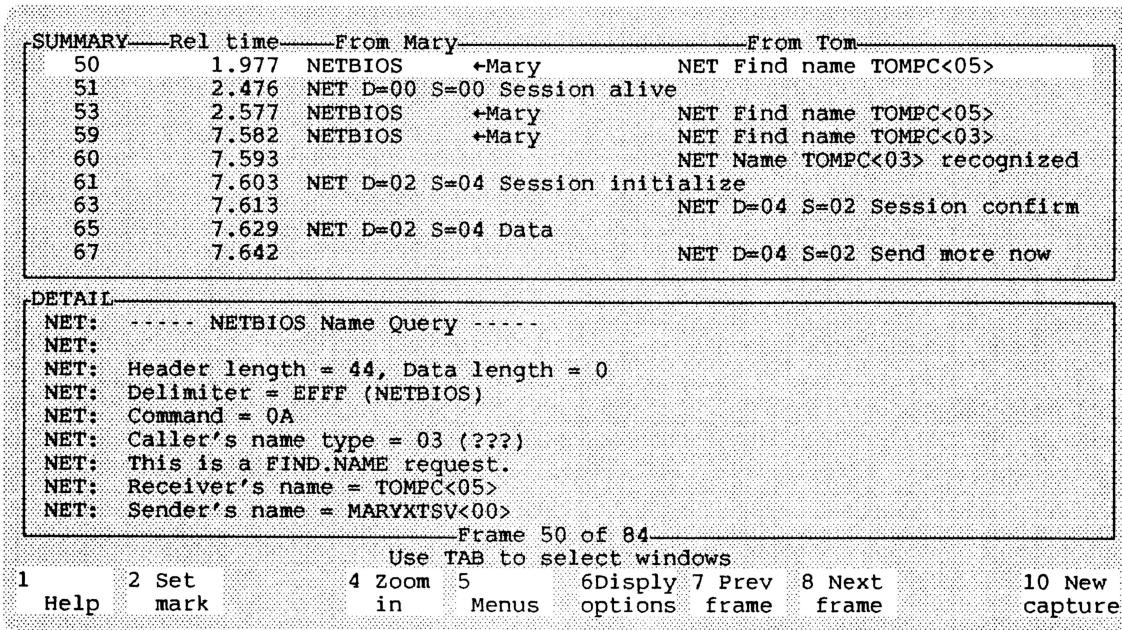


Figure 2-8: Summary and Detail views of frames in which station Mary seeks to identify station Tom in order to send a message.

## Search for Forwarded Address

In Figure 2-8, you can see the station name followed by a number enclosed in angle brackets. The NETBIOS protocol allows a total of 16 bytes for a machine's logical addresses, and permits a machine to adopt several alternate names by which it may be addressed. By convention, the first 15 bytes of these logical names are used for the name itself, and the sixteenth is reserved for a qualifier. The Sniffer's display routines show the name with trailing blanks removed, and the sixteenth byte displayed by its hexadecimal value enclosed in angle brackets. In this example, station MARYXTSV sends a broadcast message seeking a station whose name is TOMPC with the qualifier <05>.

The strategy used by the message handling software now becomes apparent. The <05> qualifier indicates an alias used to receive messages on behalf of another station. The program starts by

broadcasting a request for the forwarded name TOMPC. In effect, the machine called Mary is asking “Is there anyone out there whose name is not Tom but who is receiving messages forwarded from Tom?”

Since Tom is not forwarding messages, no one replies to the broadcast request.

The machine called Mary waits a while and then repeats the request for a machine that is accepting mail forwarded from Tom. By setting the display to Relative Time and marking the first such request, you can see the times at which the request is repeated. After the initial request, it is sent again five times at half-second intervals.

## Request for Name without Forwarding

After the sixth request has gone unanswered, at relative time 7.582, the machine Mary sends a request for the name Tom but now with the qualifier <03>, indicating a machine using “Tom” as a name for itself (without forwarding). Tom’s machine replies at once, and they set up a session to transmit the message. Figure 2-9 shows the frames that concern transmission (frames 65 to 73), and the *detail* and *hex* views for frame 65, in which the message is actually transmitted.

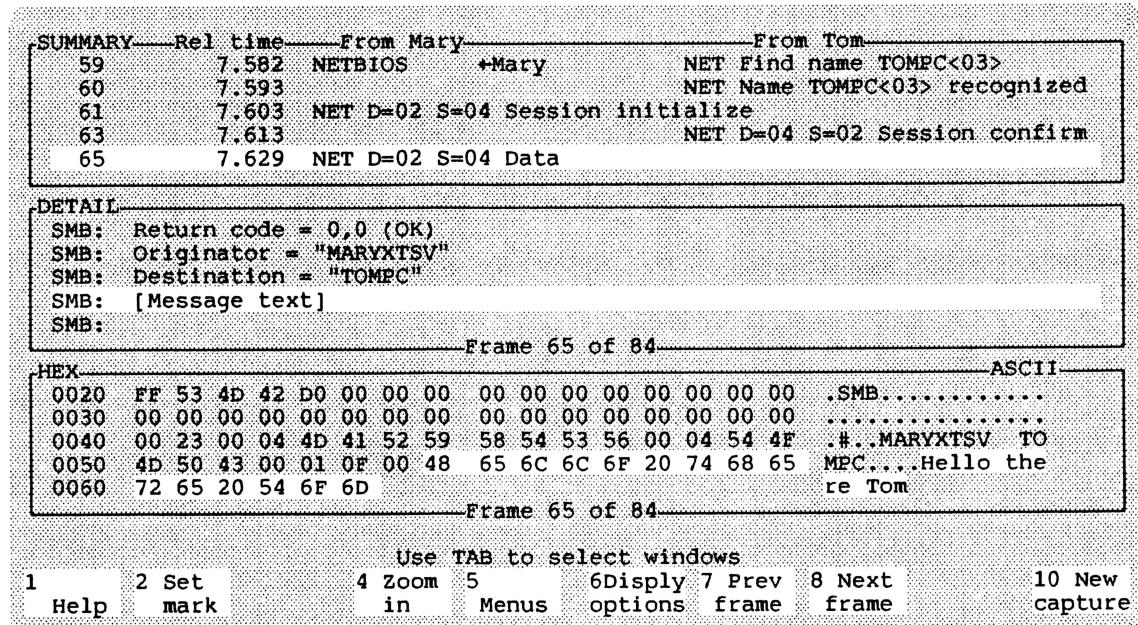


Figure 2-9: Transmission of the message.

## The Message Text

Frame 65 contains both the description of the message (its source and address) and the message text itself. Apparently the software at the receiving end notes only the description, and then requests the machine called MARYXTSV to resend the message. It seems to be unaware that it has already received the message. In frame 60, the machine called MARYXTSV sends the message again, and this time the machine called TOMPC acknowledges its receipt.

Using two viewports, you can bring the display of frame 65 and the display of frame 69 to the screen at the same time. They are in fact identical.

HEX	ASCII	HEX	ASCII
0000 18 40 10 00 5A 00 2F EB .@..Z./.		0000 18 40 10 00 5A 00 2F EB .@..Z./.	
0008 10 00 5A 00 33 BF F0 F0 ..Z.3...		0008 10 00 5A 00 33 BF F0 F0 ..Z.3...	
0010 E2 C6 OE 00 FF EF 16 01 ..!....		0010 E4 C8 OE 00 FF EF 16 00 ..!....	
0018 00 00 00 00 21 00 02 04 ..!....		0018 01 00 00 00 22 00 02 04 .."....	
0020 FF 53 4D 42 D0 00 00 00 .SMB....		0020 FF 53 4D 42 D0 00 00 00 .SMB....	
0028 00 00 00 00 00 00 00 00 ..!....		0028 00 00 00 00 00 00 00 00 ..!....	
0030 00 00 00 00 00 00 00 00 ..!....		0030 00 00 00 00 00 00 00 00 ..!....	
0038 00 00 00 00 00 00 00 00 ..!....		0038 00 00 00 00 00 00 00 00 ..!....	
0040 00 23 00 04 4D 41 52 59 #..MARY		0040 00 23 00 04 4D 41 52 59 #..MARY	
0048 58 54 53 56 00 04 54 4F XTSV..TO		0048 58 54 53 56 00 04 54 4F XTSV..TO	
0050 4D 50 43 00 01 0F 00 48 MPC...H		0050 4D 50 43 00 01 0F 00 48 MPC...H	
0058 65 6C 6C 6F 20 74 68 65 ello the		0058 65 6C 6C 6F 20 74 68 65 ello the	
0060 72 65 20 54 6F 6D re Tom		0060 72 65 20 54 6F 6D re Tom	

Use TAB to select windows

Frame 65 of 84                                  Frame 69 of 84

1 Help    2 Set mark    4 zoom in    5 Menus    6Display options    7 Prev frame    8 Next frame    10 New capture

Figure 2-10: Using two viewports to compare the frames in which the message is transmitted.

## Conclusions from the Inquiry

From these few investigations with the Sniffer, it seems reasonable to draw some preliminary conclusions for further investigation:

- The immediate cause of the delay in transmitting the message is the long interval (totaling more than 7 seconds) before the sending machine concludes that no machine is going to acknowledge its broadcast search for a forwarded address.
- A related cause is the software designer's decision routinely to seek a forwarded address before attempting to send to the address directly. One can't tell from the Sniffer data why the designer adopted that strategy, but it may bear reviewing.

- A minor redundancy arises because the receiver apparently fails to note that the message text is included in the first data transmission, and so asks (unnecessarily) to have it transmitted again. It might pay to review that point with the software designer.

## Example 2: Inserting a New Station into the Ring

### Ring Purge

A token ring network depends on each station to relay every transmission to the station downstream from it; normally, only the station that originated a transmission can strip it from the ring. Whenever a station is connected or disconnected from the ring (either by physically attaching or detaching a connector, or turning a station's power supply on or off), the flow through that point is disrupted. A message then in transit is almost certainly disrupted. The next station downstream may detect a defective transmission, or the originating station may detect that its message has failed to return to it. When no message is in transit, the free token is circulating, and it too is likely to be lost during insertion or removal of a station. The active monitor detects loss of the free token.

After any of these events, the machine playing the role of Active Monitor broadcasts a *ring purge* message, and thereby sets in motion a chain of events that recover the lost information (if any), and establish the new sequence of stations in the ring.

Because connect and disconnect are quite common, the ripple of error and recovery that accompanies each event is likely to turn up in Sniffer records from an active ring. It's instructive to note what this sequence looks like, both to see the ring in action, and to be aware of the normal pattern of such events so you distinguish them from the trouble conditions you're looking for. The example that follows records the events when one new station, whose name is here displayed as *NEW Station*, is inserted in a minimal ring which initially contains only the Sniffer and one other station, whose name is here displayed as *Already On*. In this example, the Sniffer happens to have the role of active monitor.

SUMMARY	Rel time	DST	SRC	
1	-21.823	Broadcast	+This Sniffer	MAC Active Monitor Present
2	-21.803	Broadcast	+Already On	MAC Standby Monitor Present
3	-14.896	Broadcast	+This Sniffer	MAC Active Monitor Present
4	-14.884	Broadcast	+Already On	MAC Standby Monitor Present
5	-7.968	Broadcast	+This Sniffer	MAC Active Monitor Present
6	-7.956	Broadcast	+Already On	MAC Standby Monitor Present
7	-1.041	Broadcast	+This Sniffer	MAC Active Monitor Present
8	-1.028	Broadcast	+Already On	MAC Standby Monitor Present
M	9	0.000	Broadcast	+This Sniffer
	10	0.000	NEW Station	MAC Duplicate Address Test
	11	0.000	Broadcast	+This Sniffer
	12	0.000	NEW Station	MAC Active Monitor Present
	13	0.011	Broadcast	MAC Duplicate Address Test
	14	0.011	LAN Manager	+NEW Station
	15	0.027	Broadcast	+NEW Station
	16	0.027	Param Server	+NEW Station
	17	0.028	LAN Manager	+This Sniffer
	18	0.028	Param Server	+NEW Station
	19	0.028	Param Server	+NEW Station
	20	0.029	Param Server	+NEW Station
				MAC Request Initialization
				MAC Request Initialization

1      2 Set      5      6Display      7 Prev      8 Next      10 New  
Help      mark      Menus      options      frame      frame      capture

Figure 2-11: MAC frames surrounding insertion of a new station into the ring.

Frames earlier than frame 9 show only the regular “monitor present” sequence every 7 seconds.

At frame 9, the active monitor notes an error (probably the passage of a millisecond without detecting a token), and announces a ring purge. A station which had transmitted but not heard its own transmission returned to it would then repeat it. In this example (as is likely in a small or lightly loaded network) no transmission was interrupted, and so none is re-sent.

## Duplicate Address Test

In frame 10, the new station sends a *duplicate address test* as a message addressed to itself. That’s a way of verifying that no other device on the network has the same station address hardcoded in its adapter. It repeats that in frame 12. (Meanwhile, in frame 11, the Active Monitor sends its regular “heartbeat.”)

## Change in Upstream Neighbor Address

In frame 14, the new station reports its upstream neighbor in a broadcast message addressed to the LAN Manager. This message doesn’t call for a reply. If a device on the network is acting as LAN Manager, it notes this message as a way of tracking the current network topology. In frame 17, the Sniffer makes a similar broadcast message. Examining those messages with the *detail* view shows that the station whose name we’ve displayed as *Already On* is upstream from *NEW Station*, and that *NEW Station* is upstream from the Sniffer (Figure 2-11).

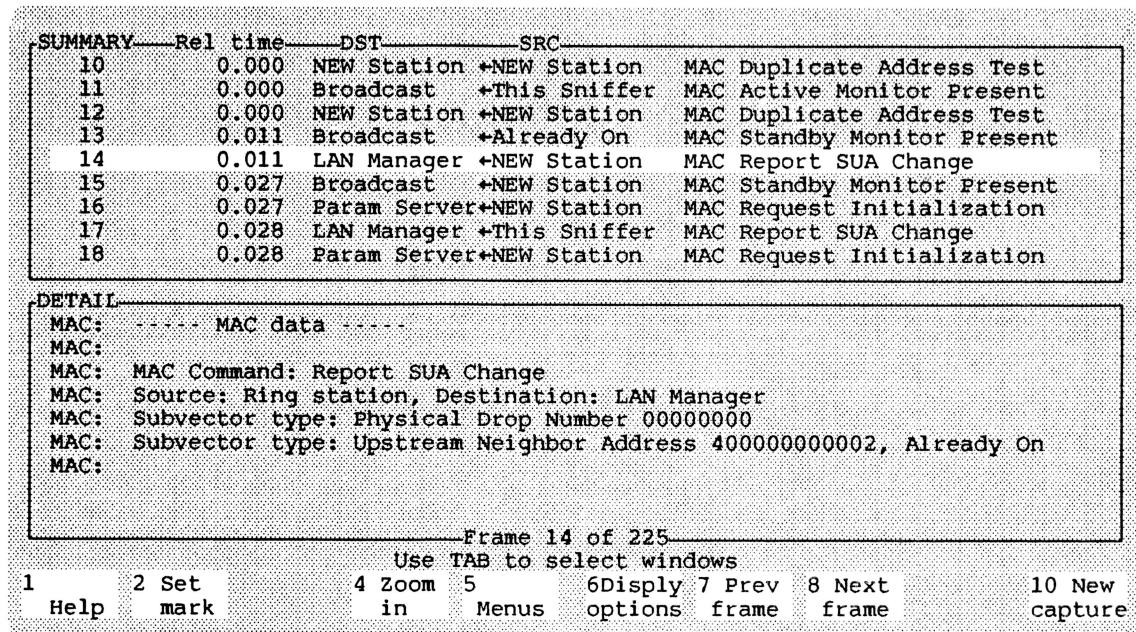


Figure 2-12: Detail view of frame showing report of upstream neighbor's address.

## Initialization Request

In frames 16, 18, 19 and 20, *NEW Station* makes four requests for initialization, directed to the station acting as Parameter Server (but gets no reply since no station is playing that role).

## Active Monitor's Final Report of the Insertion

In frame 134, almost 2 seconds after the insertion, the Active Monitor reports the soft error that it detected. The station playing the role of Error Monitor (if any is) can log the event.

### **Example 3: Batch File on a Server's Disk**

A network administrator encourages users to read programs they use regularly from storage on a central file server. That includes not only application software, but also batch files that perform a variety of minor but convenient tasks. However, for trivial tasks it seems that network response is slower than one would expect. You decide to examine the overhead in executing a batch file that is stored on a remote machine.

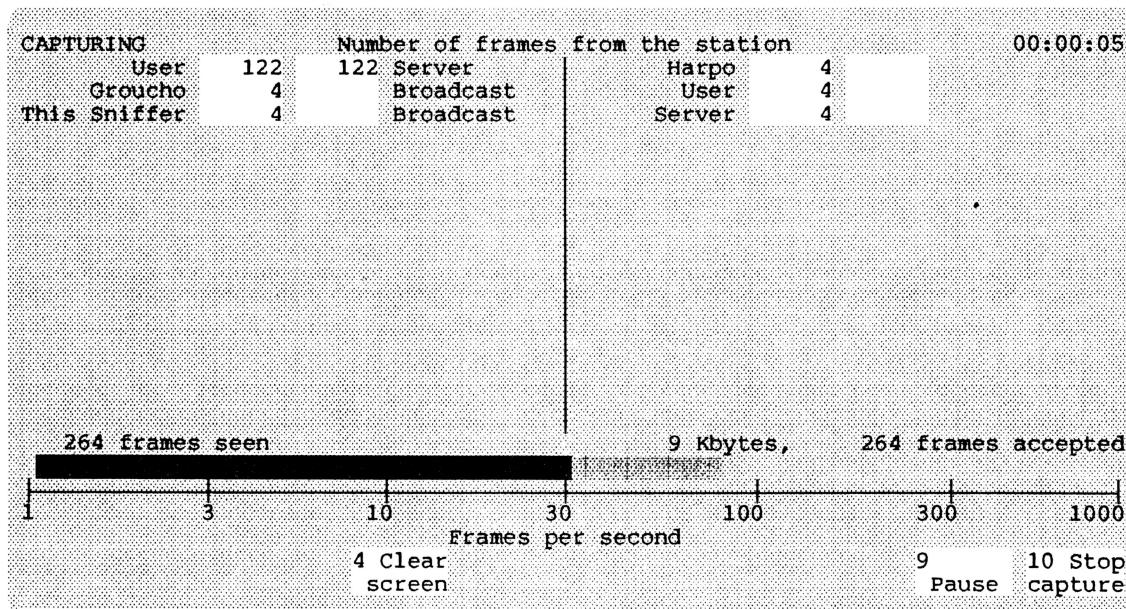
## A Test File

You arrange with a user to set up a trivial batch file so you can record the transmissions involved in executing it. The file uses only DOS built-in commands, and does almost nothing:

```
C:\> type e:test.bat
echo off
break on
echo hello there
cls
echo on
```

## Metering Data Flow

You start collecting data at the Sniffer, and have the user immediately execute the remote batch file. It's easy to see a sudden burst of activity. The counters for bytes transferred between the machines SERVER and User show 244 frames exchanged in the first few seconds of the transaction (Figure 2-13).



**Figure 2-13: Meters and counters recordings traffic between two stations during test.**

Since the volume transferred during this simple task is high, it seems worth checking further to see what's going on. You save the Capture Buffer to a file, and start examining it.

## SMB Overview

The highest level involved in the transfer is SMB. It often easiest to start at a high level to get an overview of the transaction before examining the details. Filtering for SMB frames, and showing them in two-station format gives a summary (Figure 2-14). The summary is surprisingly long, quite a bit more than will fit on a single screen. You could scroll through it. Alternatively, you could print it; in this case it all fits on a single sheet, so Figure 2-14 shows the printed form.

Frame	Rel time	From User	From Server
3	0.000	SMB C Search \TEST.???	
7	0.016		SMB R 1 entry found (done)
17	0.073	SMB C Continue search	
21	0.087		SMB R No more files
25	0.239	SMB C Open \TEST.BAT	
29	0.261		SMB R F=0000 Opened
33	0.289	SMB C F=0000 Read 512 at 0 (offset)	
37	0.303		SMB R OK
47	0.357	SMB C F=0000 Close	
51	0.369		SMB R Closed
55	0.521	SMB C Open \TEST.BAT	
59	0.543		SMB R F=0000 Opened
63	0.571	SMB C F=0000 Read 512 at 10 (offset)	
67	0.583		SMB R OK
71	0.597		SMB R OK
81	0.651	SMB C F=0000 Close	
85	0.663		SMB R Closed
89	0.781	SMB C Open \TEST.BAT	
93	0.803		SMB R F=0000 Opened
97	0.831	SMB C F=0000 Read 512 at 20 (offset)	
101	0.845		SMB R OK
111	0.901	SMB C F=0000 Close	
115	0.912		SMB R Closed
119	0.925		SMB R Closed
123	1.092	SMB C Open \TEST.BAT	
127	1.113		SMB R F=0000 Opened
131	1.143	SMB C F=0000 Read 512 at 38 (offset)	
135	1.156		SMB R OK
145	1.207	SMB C F=0000 Read 512 at 38 (offset)	
149	1.220		SMB R OK
159	1.265	SMB C F=0000 Close	
163	1.277		SMB R Closed
167	1.420	SMB C Open \TEST.BAT	
171	1.441		SMB R F=0000 Opened
175	1.471	SMB C F=0000 Read 512 at 43 (offset)	
179	1.484		SMB R OK
189	1.533	SMB C F=0000 Read 512 at 43 (offset)	
193	1.545		SMB R OK
203	1.594	SMB C F=0000 Close	
207	1.606		SMB R Closed
211	1.759	SMB C Open \TEST.BAT	
215	1.781		SMB R F=0000 Opened
219	1.810	SMB C F=0000 Read 512 at 52 (offset)	
223	1.823		SMB R OK
227	1.851	SMB C F=0000 Close	
231	1.863		SMB R Closed

Figure 2-14: Printer output, execution of a batch file, SMB frames.

## **Repeated Open and Close**

In the course of executing the five commands in TEST.BAT, the same file is opened and closed six times. There are eight SMB commands to read the five lines of data.

Evidently the network operating system requests a separate read for each line of the batch file, rather than attempting to read the entire file once. Apparently it opens and then closes the file for each line. Perhaps it elects this strategy because a single line of a batch file may launch work that takes considerable time, or may require resources that would be tied up by leaving the batch file open.

## **Questions Arising from the SMB Display**

There are curiosities to note in the sequence of SMB commands. Following them up may be instructive. Here are some:

- There are two search commands, frames 3 and 17. Why?
- There are consecutive acknowledgments without an intervening SMB command (frames 67 and 71). Why?
- Twice there are consecutive requests to read (frames 131 and 145; frames 175 and 189). This is unlike the sequence for reading other lines of the batch file. Why?
- There are two consecutive confirmations that the file has been closed (frames 115 and 119). Why?

## **Overall Communication Density**

Before looking at details, it may be worth asking the size of the effect. Is it really true that the response is slow, or the data transmission heavy in this batch file? You can measure the overall impact by calculating the elapsed time and the network utilization.

Since the SMB frames are embedded in NETBIOS and LLC frames, and additional NETBIOS and LLC frames are required to acknowledge them, you need to re-set the filter to pass both plain LLC frames and LLC frames that contain NETBIOS frames, as well as those that include SMB commands. At the same time, you should turn off the “highest-level only” option for the *summary* view so that multiple levels become visible.

The sequence of commands starts with frame 3, which transmits the first request from User to Server, and ends in frame 234, with the final acknowledgment that the Server has closed the file at the end of the job. Set the mark at frame 3 (Figure 2-15).

SUMMARY	Rel time	From User	From Server
1	2.662	LLC C D=F0 S=F0 RR NR=34 P	
2	2.662		LLC R D=F0 S=F0 RR NR=53 F
M 3	0.000	SMB C Search \TEST.???	
4	0.001		LLC R D=F0 S=F0 RR NR=54
5	0.003		LLC C D=F0 S=F0 I NR=54 NS=34
6	0.006	LLC R D=F0 S=F0 RR NR=35	
7	0.016		SMB R 1 entry found (done)
8	0.020	LLC R D=F0 S=F0 RR NR=36	
9	0.026	LLC C D=F0 S=F0 I NR=36 NS=54	
10	0.027		LLC R D=F0 S=F0 RR NR=55
11	0.037	LLC C D=F0 S=F0 I NR=36 NS=55	
12	0.039		LLC R D=F0 S=F0 RR NR=56
13	0.041		LLC C D=F0 S=F0 I NR=56 NS=36
14	0.044	LLC R D=F0 S=F0 RR NR=37	
15	0.050	LLC C D=F0 S=F0 I NR=37 NS=56	
16	0.052		LLC R D=F0 S=F0 RR NR=57
17	0.073	SMB C Continue search	
18	0.075		LLC R D=F0 S=F0 RR NR=58
19	0.077		LLC C D=F0 S=F0 I NR=58 NS=37
20	0.080	LLC R D=F0 S=F0 RR NR=38	

1 Help    2 Set mark    5 Menus    6 Display options    7 Prev frame    8 Next frame    10 New capture

Figure 2-15: LLC and SMB frames, with mark set at the start of the sequence (frame 3).

Then jump to show frame 234, displaying time relative to the marked frame (Figure 2-15).

SUMMARY	Rel time	From User	From Server
226	1.835		LLC R D=F0 S=F0 RR NR=113
227	1.851	SMB C F=0000 Close	
228	1.853		LLC R D=F0 S=F0 RR NR=114
229	1.855		LLC C D=F0 S=F0 I NR=114 NS=8
230	1.858	LLC R D=F0 S=F0 RR NR=87	
231	1.863		SMB R Closed
232	1.868	LLC R D=F0 S=F0 RR NR=88	
233	1.873	LLC C D=F0 S=F0 I NR=88 NS=114	
234	1.875		LLC R D=F0 S=F0 RR NR=115
239	5.335	LLC C D=F0 S=F0 RR NR=88 P	
240	5.335		LLC R D=F0 S=F0 RR NR=115 F

1 Help    2 Set mark    5 Menus    6 Display options    7 Prev frame    8 Next frame    10 New capture

Figure 2-16: Summary view of LLC and SMB frames (showing all levels), with time relative to the start of the batch file sequence.

As you can see in Figure 2-16, the total elapsed time is 1.875 seconds. How much of that time was devoted to the transmissions required for this task? Alter the view options to show *Network Utilization* instead of *Relative Time* (Figure 2-17).

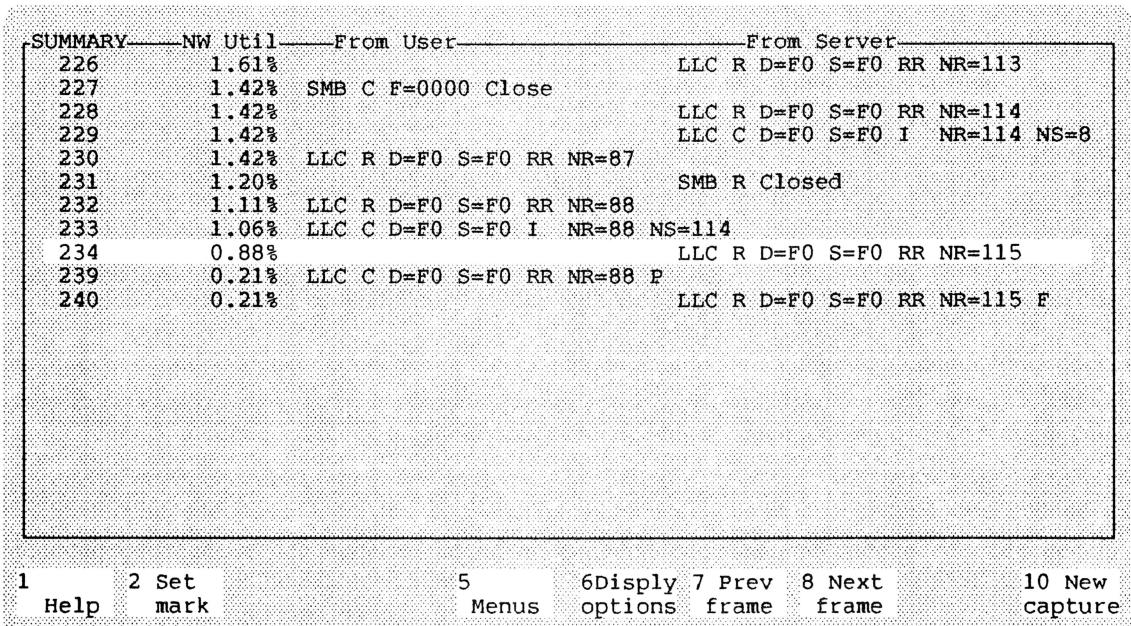


Figure 2-17: Percentage of network bandwidth utilized by the selected frames during a 100-millisecond window around each frame.

As you can see in Figure 2-17, in the time around the 23 messages of this exchange, around 1% of the network's total 4 million bits-per-second bandwidth was devoted to them. The real-time display recorded during the exchange (in the form shown in Figure 2-13) logged more than 9,000 bytes transmitted to achieve a single execution of a 52-byte batch file (in which the batch file caused no data transfer at all).

The network's bandwidth is so large that for a small number of users, this level of redundancy may well go unnoticed. But with a larger number of users, 1% of the bandwidth may become significant.

## A Detailed Look at the Search for the Batch File

The sequence starts when the user types

test

from within a network directory (or virtual drive), causing the network operating system to ask the Server machine whether it has such a file. The User machine's first step is to ask the server whether such a file exists (Figure 2-18). The request doesn't need to spell out the name of the Server's directory since that has been established earlier, and is referred to here only by its NPID (net path ID).

```

DETAIL
SMB: ----- SMB Search Directory Command -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 0,0 (OK)
SMB: File pathname = "\TEST.???"
SMB: Maximum number of search entries to return = 11
SMB: Attribute flags = 13

Frame 3 of 240

HEX ASCII
0000 10 40 10 00 5A 00 33 BF 10 00 5A 00 2F EB F0 F0 .@..z.3...z./...
0010 6A 44 0E 00 FF EF 16 00 00 00 00 00 49 00 01 02 JD.....I...
0020 FF 53 4D 42 81 00 00 00 00 00 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 00 E6 A4 C0 10 00 00 00 00 00 .....
0040 02 0B 00 13 00 0E 00 04 5C 54 45 53 54 2E 3F 3F .....TEST.??
0050 3F 00 05 00 00 ?.....

```

Frame 3 of 240

Use TAB to select windows

1	2 Set		4 Zoom	5	6Display	7 Prev	8 Next	10 New
Help	mark		in	Menus	options	frame	frame	capture

Figure 2-18: Request to search for file TEST on Server machine.

The Server reports that it has found one file matching that name, and that its name is C:TEST.BAT (Figure 2-19).

```

DETAIL
SMB: ----- SMB Search Directory Response -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 1,18 (No more files)
SMB: Entries returned = 1
SMB: ...A. .... C:TEST.BAT 52 bytes 20-Sep-86 17:57:36
SMB:

Frame 7 of 240

HEX ASCII
0000 10 40 10 00 5A 00 2F EB 10 00 5A 00 33 BF F0 F0 .@..Z./...Z.3...
0010 46 6C 0E 00 FF EF 16 01 00 00 00 00 5D 00 02 01 Fl.....]...
0020 FF 53 4D 42 81 01 00 12 00 80 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 00 E6 A4 C0 10 00 00 00 00 00 .....
0040 01 01 00 2E 00 01 2B 00 03 54 45 53 54 20 20 20 .....+..TEST
0050 20 3F 3F 3F 13 0C 00 00 00 00 00 00 00 00 00 20 32 8F ???.2.
0060 34 0D 34 00 00 00 54 45 53 54 2E 42 41 54 00 42 4.4...TEST.BAT.B
0070 41 54 00 AT.

Frame 7 of 240
Use TAB to select windows

```

1	2 Set		4 Zoom	5	6Display	7 Prev	8 Next	10 New
Help	mark		in	Menus	options	frame	frame	capture

Figure 2-19: Server's reply to the search request.

However, although we at the Sniffer can see in Frame 7 that the Server has sent a reply stating the number of matching file names found and listing what they were, the User machine is not ready to receive the information. It sends the NETBIOS "No Receive" message (Figure 2-20), and indicates that it was able to accept

only the first 40 bytes of the message. Those 40 bytes are the SMB header, so it has not yet received any of the SMB reply.

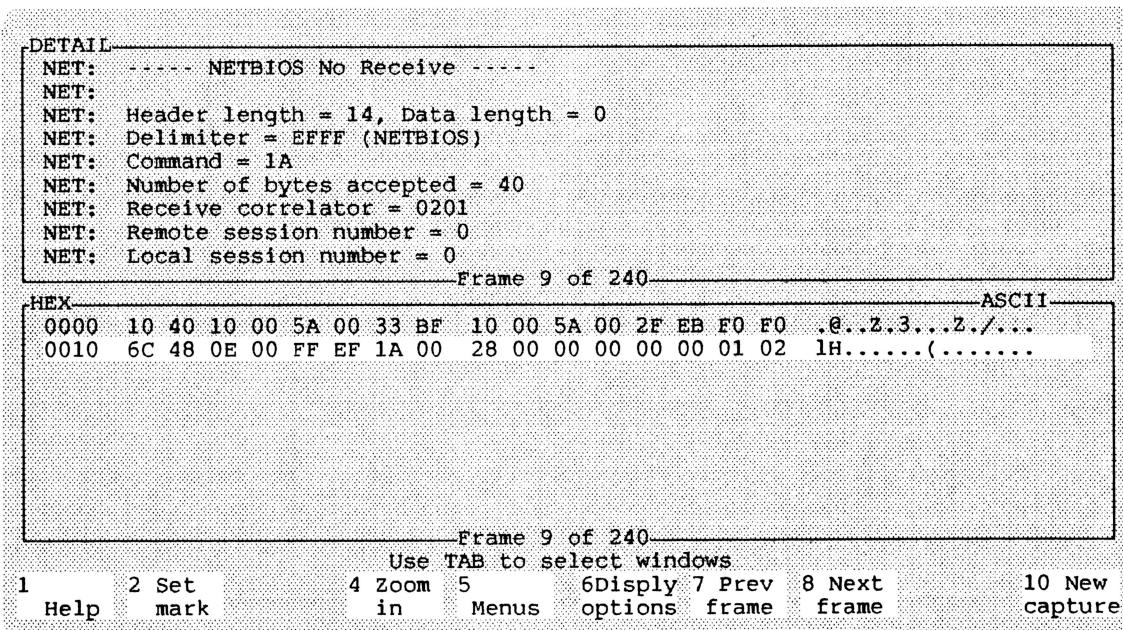


Figure 2-20. User machine says it was unable to receive.

When the User machine is ready to receive the rest of the reply from the Server, it sends the SMB command "Receive outstanding" (Figure 2-21).

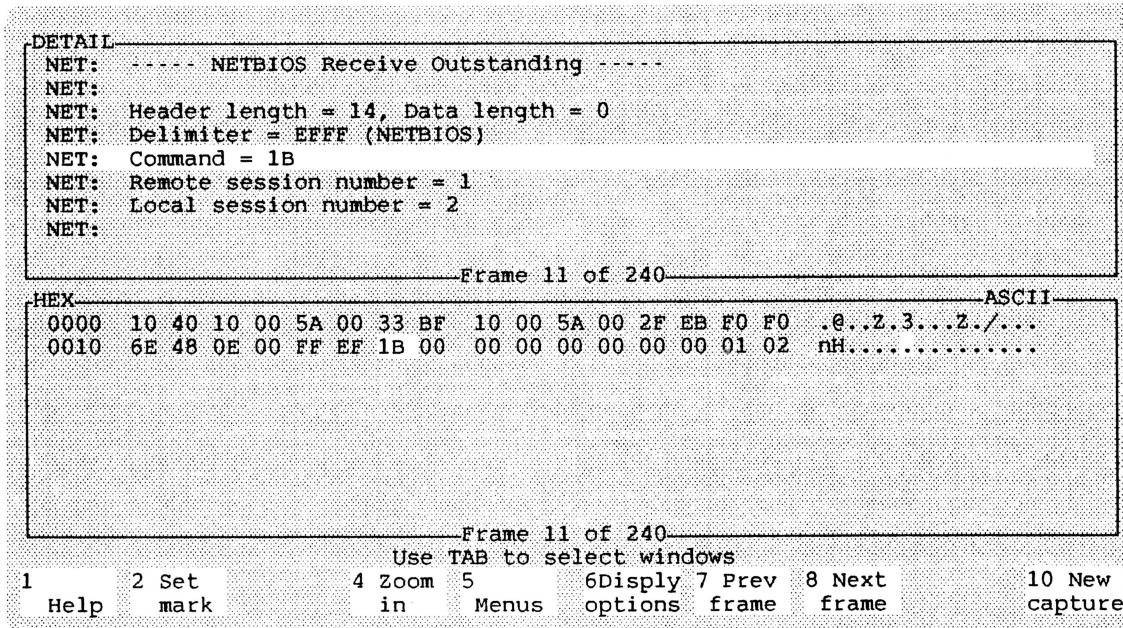


Figure 2-21: The User machine says it is now ready to receive the balance of the transmission.

The Server sends the remainder of the message, which in fact contains the entire SMB message, since none of it got through the first time (Figure 2-21).

```

DETAIL
NET: ----- NETBIOS Data Only Last -----
NET:
NET: Header length = 14, Data length = 43
NET: Delimiter = EFFF (NETBIOS)
NET: Command = 16
NET: Transmit correlator = 0000
NET: Remote session number = 2
NET: Local session number = 1
NET:

Frame 13 of 240
HEX ASCII
0000 10 40 10 00 5A 00 2F EB 10 00 5A 00 33 BF F0 F0 .E..Z./...Z.3...
0010 48 70 0E 00 FF EF 16 00 01 00 00 00 5E 00 02 01 H.....?.....
0020 03 54 45 53 54 20 20 20 20 3F 3F 13 OC 00 00 .TEST ???.?...
0030 00 00 00 00 00 20 32 8F 34 0D 34 00 00 00 54 45 .....2.4.4...TE
0040 53 54 2E 42 41 54 00 42 41 54 00 ST.BAT.BAT.

Frame 13 of 240
Use TAB to select windows
1 Help 2 Set mark 4 Zoom in 5 Menus 6Display options 7 Prev frame 8 Next frame 10 New capture

```

Figure 2-22: Server repeats the name of the file it has found.

```

DETAIL
SMB: ----- SMB Search Directory Command -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 0,0 (OK)
SMB: Continuation search
SMB: [Search parameter control block]
SMB:

Frame 17 of 240
HEX ASCII
0000 10 40 10 00 5A 00 33 BF 10 00 5A 00 2F EB F0 F0 .E..Z.3...Z./...
0010 72 4A 0E 00 FF EF 16 00 00 00 00 00 4A 00 01 02 rJ.....J....
0020 FF 53 4D 42 81 00 00 00 00 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 B6 A4 C0 10 00 00 00 00 .....
0040 02 0B 00 2E 00 1A 00 04 00 05 15 00 03 54 45 53 .....TES
0050 54 20 20 20 20 3F 3F 13 OC 00 00 00 A2 00 3B T ???....;
0060 10 .

Frame 17 of 240
Use TAB to select windows
1 Help 2 Set mark 4 Zoom in 5 Menus 6Display options 7 Prev frame 8 Next frame 10 New capture

```

Figure 2-23: The User machine requests continuation of the search for the file TEST.

Frame 17 (shown in Figure 2-23) seems to reflect a misunderstanding. The Server has already reported that it

completed its search for TEST, that it found 1 match, and that the name was C:TEST.BAT. The User machine appears to ignore the information that the search has been completed, and asks to continue the search. The Server replies that it has no more (Figure 2-24).

```

DETAIL
SMB: ----- SMB Search Directory Response -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 1,18 (No more files)
SMB: Entries returned = 0
SMB:
SMB: [Normal end of "SMB Search Directory Response" packet.]
Frame 21 of 240

HEX ASCII
0000 10 40 10 00 5A 00 2F EB 10 00 5A 00 33 BF F0 F0 .e...z./...z.3...
0010 4C 74 0E 00 FF EF 16 01 00 00 00 00 5F 00 02 01 Lt.....
0020 FF 53 4D 42 81 01 00 12 00 80 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 00 E6 A4 C0 10 00 00 00 00 .....
0040 01 00 00 03 00 01 00 00

Frame 21 of 240
Use TAB to select windows
1 Help 2 Set mark 4 Zoom in 5 Menus 6 Dispaly options 7 Prev frame 8 Next frame 10 New capture

```

Figure 2-24: Server says it can no find no more entries for TEST.???.

## Reading the First Line of the Batch File

As noted earlier, the User machine's strategy is to open the file, read a line of the batch file, close the file, execute that line, and then repeat. So you look more closely at how it reads the first line of the file.

Frame 25 contains the SMB command to open the file. It's shown in Figure 2-25.

```

DETAIL
SMB: Return code = 0,0 (OK)
SMB: File pathname = "\TEST.BAT"
SMB: Access flags = 00
SMB:     0... .... = Pass access to any sub-processes
SMB:     .000 .... = MS-DOS compatibility exclusive open
SMB:     .... 0000 = Open file for reading
SMB: Attribute flags = 16
SMB:     00.. .... = Reserved
SMB:     ..0. .... = File(s) not changed since last archive
Frame 25 of 240

HEX
0000 10 40 10 00 5A 00 33 BF 10 00 5A 00 2F EB F0 F0 .@..Z.3...Z./...
0010 76 4E 0E 00 FF EF 16 00 00 00 00 00 4B 00 01 02 vN.....K....
0020 FF 53 4D 42 02 00 00 00 00 00 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 00 E6 A4 C0 10 00 00 00 00 00 .....
0040 02 00 00 16 00 0B 00 04 5C 54 45 53 54 2E 42 41 .....\\TEST.BA
0050 54 00 T.

ASCII
Frame 25 of 240
Use TAB to select windows
1 Help   2 Set mark   4 Zoom in   5 Menus   6Display options   7 Prev frame   8 Next frame   10 New capture

```

Figure 2-25: The User machine's request to open the batch file.

In due course the Server replies that it has opened the file, and supplies the file handle. It also reports the date and time at which the file was created, and its total size in bytes (Figure 2-25).

```

DETAIL
SMB: ----- SMB Open File Response -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 0,0 (OK)
SMB: File handle = 0000
SMB: Attribute flags = 20
SMB:     00.. .... = Reserved
SMB:     ..1. .... = File(s) changed and not archived
SMB:     ...0 .... = No directory file(s)
SMB:     .... 0... = No volume label info
SMB:     .... 0.. = No system file(s)
SMB:     .... 0. = No hidden file(s)
SMB:     .... .0 = No read only file(s)
SMB: Creation date = 19-Sep-86 17:57:36
SMB: File size = 52
SMB: Access flags = 02
SMB:     0.... .... = Pass access to any sub-processes
SMB:     .000 .... = MS-DOS compatibility exclusive open
Frame 29 of 240

1 Help   2 Set mark   5 Menus   6Display options   7 Prev frame   8 Next frame   10 New capture

```

Figure 2-26: Display of frame 29, zoomed so the Detail view has the entire screen, showing the Server's reply to the request to open file TEST.BAT.

The Server machine undertakes to read the first line of the file. It's keeping track of how much it has read, so it stipulates that reading should start at offset 0. Curiously, it asks to read 512

bytes, although it has received a message informing it that the size of the file is 52 bytes (Figure 2-27).

```

DETAIL
SMB: ----- SMB Read Byte Range Command -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 0,0 (OK)
SMB: File handle = 0000
SMB: Number of bytes to read = 512
SMB: Offset in file = 0
SMB: Total (estimated) read count = 512
SMB:
SMB: [Normal end of "SMB Read Byte Range Command" packet.]
SMB:

Frame 33 of 240
1 Help 2 Set mark 5 Menus 6 Dispaly options 7 Prev frame 8 Next frame 10 New capture

```

Figure 2-27: The User machine asks for the first 512 bytes of the file TEST.

In response, the Server returns all 52 bytes, in a message marked “data final” to indicate that there isn’t more to come (Figure 2-28).

```

DETAIL
SMB: ----- SMB Read Byte Range Response -----
SMB:
SMB: Net path (NPID) = A4E6
SMB: Process id (PID) = 10C0
SMB:
SMB: Return code = 0,0 (OK)
SMB: Number of bytes read = 52
SMB: [52 bytes of read data]
SMB:

Frame 37 of 240
HEX ASCII
0000 10 40 10 00 5A 00 2F EB 10 00 5A 00 33 BF F0 F0 .@...z./...z.3...
0010 54 7C 0E 00 FF EF 16 01 00 00 00 00 61 00 02 01 T|.....a...
0020 FF 53 4D 42 0A 00 00 00 00 80 00 00 00 00 00 00 .SMB.....
0030 00 00 00 00 00 00 00 E6 A4 C0 10 00 00 00 00 00 .....
0040 05 34 00 00 00 00 00 00 00 00 00 37 00 01 34 00 .4,...,7..4.
0050 65 63 68 6F 20 6F 66 66 0D 0A 62 72 65 61 6B 20 echo off..break
0060 6F 6E 0D 0A 65 63 68 6F 20 68 65 6C 6C 6F 20 74 on..echo hello t
0070 68 65 72 65 0D 0A 63 6C 73 0D 0A 65 63 68 6F 20 here..cls..echo
0080 6F 6E 0D 0A on..

Frame 37 of 240
Use TAB to select windows
1 Help 2 Set mark 4 Zoom in 5 Menus 6 Dispaly options 7 Prev frame 8 Next frame 10 New capture

```

Figure 2-28: Server transmits all 52 bytes of file TEST.BAT.

## Play it Again, Sam

Although the Server machine requested 512 bytes and said it expected to receive 512 bytes, in fact it is not ready, and sends a NETBIOS reply saying that it has been able to accept only the first 48 bytes of the transmission, which contain the header for the SMB message, but none of the file data itself (Figure 2-29).

```
DETAIL
NET: ----- NETBIOS No Receive -----
NET:
NET: Header length = 14, Data length = 0
NET: Delimiter = EFFF (NETBIOS)
NET: Command = 1A
NET: Number of bytes accepted = 48
NET: Receive correlator = 0201
NET: Remote session number = 0
NET: Local session number = 0
NET:

Frame 39 of 240
1 Help 2 Set      5 Menus 6Display 7 Prev 8 Next 10 New
      mark          options frame frame capture
```

Figure 2-29: The User machine is unable to accept the transmission.

Almost immediately the User machine says it is now ready to receive the rest of the transmission (Figure 2-30).

```
DETAIL
NET: ----- NETBIOS Receive Outstanding -----
NET:
NET: Header length = 14, Data length = 0
NET: Delimiter = EFFF (NETBIOS)
NET: Command = 1B
NET: Remote session number = 1
NET: Local session number = 2
NET:

Frame 41 of 240
1 Help 2 Set      5 Menus 6Display 7 Prev 8 Next 10 New
      mark          options frame frame capture
```

Figure 2-30: The User machine is ready for the rest of the file TEST.BAT.

In response, the Server again transmits the entire 52 bytes of the file TEST.BAT. (Figure 2-31):

The screenshot shows a NetworkMiner capture window. At the top, there's a 'DETAIL' section with various network parameters like NETBIOS Data Only Last, Header length, Delimiter, Command, Transmit correlator, Remote session number, Local session number, and a timestamp of 2000-01-01 10:00:00. Below this is a 'Frame 43 of 240' section. The frame details show a sequence of bytes from 0000 to 0050, followed by their ASCII representation. The ASCII output includes commands like 'z', '3...', 'V', 'b...', 'echo off..break', 'on..echo hello t', 'here..cls..echo on..'. At the bottom of the frame details, it says 'Use TAB to select windows'. A menu bar at the very bottom includes options 1 through 10, such as Help, Set mark, Zoom in, Menus, Display options, Prev frame, Next frame, and New capture.

Frame 43 of 240																	
NET:	----- NETBIOS Data Only Last -----																
NET:																	
NET:	Header length = 14, Data length = 52																
NET:	Delimiter = EFFF (NETBIOS)																
NET:	Command = 16																
NET:	Transmit correlator = 0000																
NET:	Remote session number = 2																
NET:	Local session number = 1																
NET:																	
Frame 43 of 240																	
HEX	ASCII																
0000	10	40	10	00	5A	00	2F	EB	10	00	5A	00	33	BF	F0	F0	.0...z./....z.3...
0010	56	80	0E	00	FF	EF	16	00	01	00	00	00	62	00	02	01	V.....b...
0020	65	63	68	6F	20	6F	66	66	0D	0A	62	72	65	61	6B	20	echo off..break
0030	6F	6E	0D	0A	65	63	68	6F	20	68	65	6C	6C	6F	20	74	on..echo hello t
0040	68	65	72	65	0D	0A	63	6C	73	0D	0A	65	63	68	6F	20	here..cls..echo
0050	6F	6E	0D	0A													on..
Frame 43 of 240																	
Use TAB to select windows																	
1	2 Set	4 Zoom	5	6 Dispaly	7 Prev	8 Next	10 New										capture
Help	mark	in	Menus	options	frame	frame											

Figure 2-31: The Server repeats transmission of the file TEST.BAT.

## Questions Raised

We've looked in detail at only two steps in the process: verifying that the requested file exists, and transmitting the first line. The cycle just shown for reading the remaining lines is more or less the same: each time, the user machine asks to open the file, and then to read 512 bytes. At each request, it calculates the offset at which it wants to start, moving past the lines it has already read. For each line, it asks to receive 512 bytes, even though the Server has just reported that the total size of the file is 52. At each line, even though it has asked for 512 bytes, it is at first unable to accept the 52 bytes transmitted, and must ask for a second transmission. For each line, it again closes the file.

The initial summary of the SMB frames showed a few other curiosities. For example, frame 119 repeats frame 115, apparently because the User machine was at first unable to receive the acknowledgment that the file had been closed.

Frame 145 repeats the read request made in frame 131. Frame 175 asks for the file starting at offset 43. Frame 181 (as usual) shows the User machine unable to accept it, and frame 183 shows the Server transmitting it again. Frame 183 shows the User machine acknowledging receipt. But frame 185 is a request from the user again to read at offset 43. Frame 189 repeats the read request made in frame 175. All told, the Server transmits the last line of TEST.BAT a total of 16 times in order to permit the User machine to execute it once.

The Sniffer offers no clue to the reasons for these apparently redundant requests-- but it does reveal that they're there.



# Chapter 3. Setting Up the Sniffer

This is a short chapter. There isn't much to do.

## Unpacking

- Remove from the carton the Sniffer, the related items packed with it, the packing list and the license agreement.
- Read the license agreement. If you feel you can't accept its terms, go no further! You have three days to put everything back in the box and return the Sniffer. When you connect the Sniffer to a power supply, by so doing you signal your acceptance of the terms of the license agreement.
- Use the packing list to check that you have everything.
- Fill out the warranty registration card so you can return it to Network General. Turn to Appendix H (at the back of this manual) and write in the serial number of your Sniffer. Appendix H is a list of points to check when you phone for help. If you write in the serial number now, you'll have it when you need it.
- Open the door of the floppy disk drive and remove the stiff card protector inserted there for shipping. Do *not* turn on power while this card is in place.

## Hardware

The Sniffer is a portable computer manufactured for Network General by COMPAQ Computer Corporation; it operates under COMPAQ DOS version 3.10. In addition to the manual you're now reading, the following reference publications are included with the Sniffer:

- *COMPAQ Portable II Personal Computer Operations Guide.*
- *MS-DOS/Basic version 3.*

The machine is equipped with:

- 8Mhz 80286 processor;
- 640 Kb main memory;
- One 20 Mb hard disk;
- One 360 Kb floppy drive;

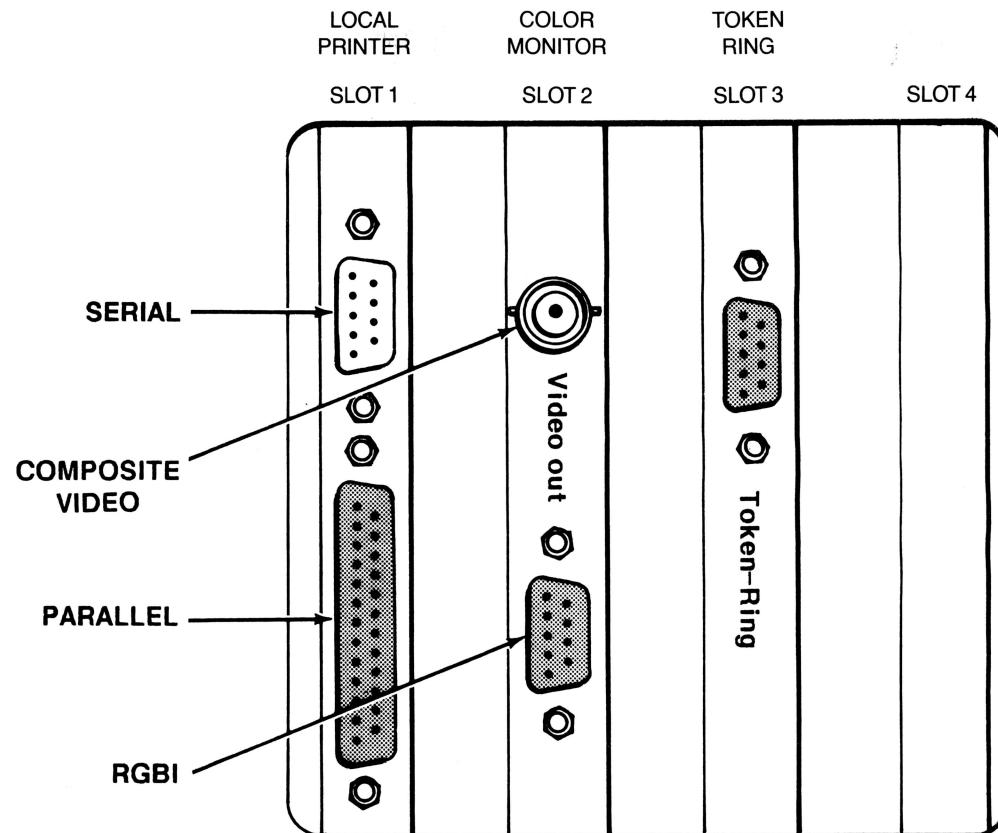
- Built-in 9-inch green monochrome monitor;
- Network adapter PCB for the IBM token ring, with its own processor and buffer RAM.
- Optionally (at extra charge), automatic switching dual-voltage power supply.

For carrying, the keyboard doubles as part of the case, and attaches so that it covers the built-in screen. A sliding cover on the right side opens to reveal the connectors to the Sniffer's adapter cards. A similar sliding cover on the left side gives access to the on-off switch and to the plug to which you connect the power cord.

## Cables

The Sniffer is supplied with two cables:

- **Power cord.** Three-prong, for 117V 60 cycles AC.
- **Token ring connector.** This 8-ft cable has an IBM token-ring connector at one end and a male DB-9 connector at the other. You plug the DB-9 connector to the corresponding plug on the card in the Sniffer's slot 3 (see Figure 3-1)



*Figure 3-1: Connections to the Sniffer's adapter cards. Note that there are two female DB-9 connectors. The one beside an RCA connector on card 2 is for the color monitor; the one alone on card 3 is for the token-ring connector.*

## Color Monitor Option

The Sniffer provides support for a color monitor. There are both advantages and disadvantages of using color. The Sniffer's display routines make use of color to highlight options selected or mark parts of the display. However, the resolution of characters (to the IBM CGA standard) is not as high as the resolution of the monochrome screen.

A color monitor is not included with the Sniffer; you have to provide your own. If you attach one, you'll get both the color display and a somewhat degraded version of the Sniffer's built-in monochrome display at the same time.

The color monitor connectors are mounted on the card in Slot 2. The concentric RCA socket is for composite video, and the DB-9 connector below it for RGBI.

**Watch out:** there are two female DB-9 connectors. The one beside an RCA connector on card 2 is for the color monitor; the one alone on card 3 is for the token-ring connector. (See figure 3-1.)

## Software

The Sniffer's software is already installed on its hard disk. There are no diskettes supplied with it.

## Starting the Sniffer

Plug it in. Turn it on. (If you're using a separate color monitor, you may need to turn it on too.)

The Sniffer program starts at once. Its first act is to ask whether you are using a color monitor. (Later, you can adjust the program so it doesn't ask again each time you start the machine.)

If you have a color monitor plugged in, answer Y for yes. Otherwise, answer N.

That takes you to an initialization screens and then to the Sniffer's main menu, from which you start each of the Sniffer's principal activities, and to which you eventually return to exit from the Sniffer program.

The last part of this chapter concerns the main menu and its conventions. From there, the work you do with the Sniffer falls into two broad classes, each of which has a chapter devoted to it:

- Chapter 4: Capturing and saving data.
- Chapter 5: Displaying and analyzing data.

However, when you first set up the Sniffer, before you start capturing or analyzing data, there are some preliminaries to take care of.

## **Color, Resolution, and Brightness**

When you're using the color monitor, characters sent to the color screen are also displayed on the Sniffer's built-in monochrome monitor. However, color characters have a lower resolution than those generated expressly for monochrome. When you answer Yes to the question about color, you get the lower-resolution characters that are characteristic of the IBM CGA standard. You get them both at the color monitor *and* at the built-in monitor. When you are using a color monitor, you probably won't want a low-resolution monochrome display at the same time. To avoid a distracting duplicate display, you can reduce the brightness of the built-in display until it's invisible.

**Watch out:** The brightness control of the Sniffer's built-in green monitor is the wheel at the bottom right of the screen, just below the disk drive. When turned to minimum, the brightness is so low that you can't see anything. When you are *not* using an external color display but the built-in screen seems blank, it may be because the brightness has been set to its minimum position. Try increasing the brightness.

## **First Time Precautions**

There are two precautions you should take before you start working with the Sniffer:

- Make a backup copy of the software, as a protection against the possibility that the pre-installed version on the hard disk is somehow damaged or lost.
- Make a system diskette with which you could start the Sniffer from its floppy drive, as a protection against the possibility that the hard disk becomes somehow unusable.

To carry out either of these procedures, you have to leave the Sniffer program and return to DOS. When you turn on the Sniffer for the first time, the first thing you'll see is the Sniffer's main menu, and the first thing you should do is move the cursor to *Exit to DOS*, and press *Enter*. Then you can carry out the two precautionary procedures described in the paragraphs that follow.

## **Organization of Software on the Hard Disk**

The software on the hard disk is organized into three principal directories:

- DOS, containing all the files used by the Sniffer's operating system other than COMMAND.COM and the two invisible files the operating system uses.
- TRSNIFF, containing TRSNIFF.EXE, the single program which manages all the Sniffer's token-ring activities. This directory contains a subdirectory called HELP, containing help files used by TRSNIFF.
- CAPTURE, a directory to contain files of captured data as you save them from the Sniffer. This directory exists, but is initially empty. (You can if you wish create additional directories for the storage of captured frames; see *Several Directories for Capture Files*, below.)
- The root directory contains the DOS file COMMAND.COM and the operating system's hidden files. It also contains AUTOEXEC.BAT, which is executed automatically whenever you turn on power or reset the machine by pressing Ctrl-Alt-Del.

## **The Autoexec File**

The file AUTOEXEC.BAT does the following automatically when the machine is started or reset:

- Sets the DOS environment so that DOS can find programs in the directories SNIFFER and DOS, and so the Sniffer can find its *names* file and its *help* file.
- Changes the current directory to CAPTURE, ready to receive files that you may save from the Capture Buffer.
- Starts the Sniffer by the file TRSTART.BAT, which asks whether you're using an external color monitor, and invokes the Sniffer by the command TRSNIFF or TRSNIFF COLOR, as appropriate.
- Following a normal exit from the Sniffer, returns you to the root directory.

## The Sniffer's Directory Conventions

The Sniffer program resides in the directory \TRSNIFF, and its help files always reside in \TRSNIFF\HELP.

It is not necessary (and not desirable) to make TRSNIFF the current directory before you start the Sniffer. The current directory should be the directory in which you expect to save files of captured data, or from which you expect to load files of captured data.

The Sniffer uses the *current* directory for three things:

- The initial directory in which to store files of captured frames from the Capture Buffer.
- The initial directory in which to look for stored files of captured frames, to load them to the Capture Buffer for analysis.
- The default directory in which to look for the names file, called STARTUP.TRD. The names file specifies the characters to appear in displays to replace or augment the hardware station addresses that the frames actually contain.

## Several Directories for Capture Files

To separate various sets of captured data (for example, data collected from different networks, or under different circumstances), you may want to set up different directories to contain the files. When the Sniffer starts work, its autoexec file makes \CAPTURE the current directory.

To instruct the Sniffer to use a different directory for saving or loading files, proceed as follows:

- From the Sniffer's main menu, select the option *Files* and then *Change path*.
- When the Sniffer displays a form showing the current path for load and save commands, edit the display so that it contains the path you want.

To make the Sniffer start automatically in a directory other than \CAPTURE, edit the file AUTOEXEC.BAT, substituting the appropriate directory name where the file formerly said CD \CAPTURE.

## **Several Names Files**

The Sniffer's display routines refer to the file STARTUP.TRD to find names to replace or augment the hardware addresses it finds in the frames it displays. To find STARTUP.TRD, the Sniffer program first looks in the current directory. If it doesn't find a name file there, it consults the environment string for the name of a directory in which to look next. In the autoexec file, the environment string is set by the command

```
SET TRNAMES = \TRSNIFF
```

When you have several different directories of saved capture files, each directory may (if you wish) have its own STARTUP.TRD. For a directory that does *not* contain its own names file, the program looks for a names file in the directory specified by the SET TRNAMES command.

## **Updating the Current Names File**

While you're setting up capture or display filters, you're permitted to refer to stations by the symbolic names they have in the current names file. If you refer to a name that isn't defined in the names file, you may have the Sniffer update the file to include the new name. (See the discussion of *Filtering by Station Address* in Chapter 4.)

## **Loading or Saving Files from Another Directory**

In the Files portion of the main menu, there's an option to set a path for files that you are going to load or save. (Its use is described in Chapter 5.)

This option in effect supplies an automatic prefix to any file names that you use. For example, if the current directory is \CAPTURE, and you specify the prefix DATA, the Sniffer loads or saves files in \CAPTURE\DATA.

However, when the prefix you provide starts with a slash (for example, \DATA) the leading slash indicates that you're not just tacking on an amendment to the DOS default directory, but providing an entire new path. In that case, the Sniffer ignores the DOS current directory and uses the path you've written instead.

*Note:* By providing a path for the Sniffer when it saves or loads capture files, you do not change the DOS "current directory," and you do not change where the Sniffer looks for a names file.

**Backing Up the Software** It is highly desirable to make for yourself the following diskettes:

- A backup copy of the software to guard against accidental erasure or damage;
- A system diskette so that you can start the machine and run the Sniffer from a diskette in the event that something prevents you from using the hard disk.

To do either of those, you must be at the DOS prompt. Since the Sniffer starts automatically, you should *select Exit* from the Sniffer's main menu. That returns you to the DOS prompt.

You can make a complete backup of everything to diskettes by typing the DOS command

```
BACKUP C:\*.* A: /S
```

It takes four diskettes to backup everything originally supplied with the Sniffer, including both the TRSNIFF and the DOS directories. (The parameter /S tells DOS to include all files in subdirectories as well.)

BACKUP prompts you for diskettes that are already formatted. It won't let you format them as needed as it proceeds. So it's important to have enough formatted diskettes on hand before you start backing up.

To limit your backup to a particular directory of captured files (for example, CAPTURE), the command is:

```
BACKUP C:\CAPTURE\*.* A:
```

You should consult the COMPAQ DOS manual for further details on the BACKUP command, or its inverse RESTORE.

## Making a Bootable Sniffer Diskette

In the event of a failure of the hard drive, it is possible to operate the Sniffer from a floppy diskette in drive A-- provided you have already prepared a suitable diskette. It is prudent to prepare such a diskette as soon as you receive the Sniffer. To do so:

- Turn on the Sniffer machine. The Sniffer program starts automatically.
- Exit to DOS.
- At the DOS prompt, type MAKEFLOP.

The program MAKEFLOP prompts you to insert a diskette in drive A, and to press *Enter* when you're ready. The program

formats the disk you supply, so it doesn't need to be formatted in advance. The format routine reports the number of bytes available on the disk, and then asks whether you want to format another. Decline this offer.

The MAKEFLOP program then transfers the needed files. When the transfer is complete, you have a bootable diskette containing all files directly used by the Sniffer, a minimum subset of DOS, and no data files.

To boot from that diskette, simply insert it in the floppy disk drive *before* you turn on power to the Sniffer or press Ctrl-Alt-Del to reset the Sniffer. When you're working from the diskette, you'll probably need a second diskette to contain files of captured frames. After you've started the Sniffer you can remove its diskette and insert a different one for data files. (If you make use of the HELP files while running from the diskette, you'll need to have the diskette actually in the drive whenever you press **F1**, since the Sniffer doesn't retain the HELP files in main memory.)

## The Sniffer's Main Menu

Whenever the Sniffer software starts, it displays the main menu. No matter what you'll be doing with the Sniffer, you always start from the main menu. You always terminate and work with the sniffer by returning to the main menu, and selecting the option **Exit**.

Figure 3-2 shows the root of the main menu.

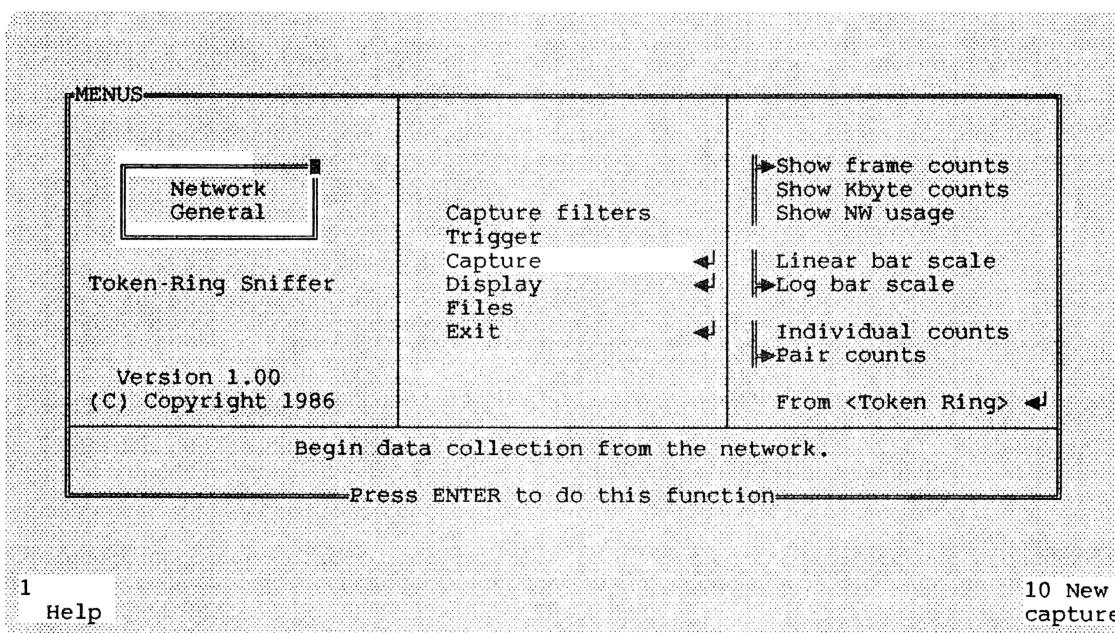


Figure 3-2: The first panel of the Sniffer's Main Menu.

## A Tree-Structured Menu

The main menu is organized as a tree. Your current position on the tree is highlighted in the center of the middle window. You traverse the tree by using the arrow keys. Each press of an arrow takes you to the next node in the direction of the arrow.

The screen shows you three panels, arranged from left to right. Immediately to the left of your current (highlighted) position is the node you just came from. Above and below you in the center panel are alternative nodes that are also reachable from the node to your left. To pick one of the alternatives, press the up arrow to move to an alternative shown higher in the center panel, or the down arrow to move to an alternative shown lower in the center panel.

Immediately to your left is the next node you'd come to if you move from your present position back towards the root of the tree. Above and below it are other nodes you'd could reach, provided you first moved one step leftward, towards the root.

To your right are nodes reachable from the node you're now on. Moving to the right takes you further out along the branch you're already on. The right panel shows you the choices you'll have if you move one step to the right, further out towards the leaves. (As you move rightward towards the leaves, the panel on the left shows where you just came from.)

Moving to the right makes available the list of choices reachable from your present position. You can already see some of them in the panel to the right. When you press the right arrow, you bring the center panel over them (so they appear in your center panel). Then you can press the up or down arrows to select the one you want. When the list of alternatives is longer than will fit in the panel at once, the others come into view as you press the up or down arrows.

To jump directly to one of the items listed in the menu's center panel, type a letter on the keyboard. The highlight moves to the next item starting with that letter.

## A Movable Viewport

As you move around the tree, your viewport moves over the tree so that your present location is always centered in the middle panel.

When you've moved all the way to the end of a branch, the panel to the right is empty, indicating that there's nowhere else to go in that direction.

## The Initial Menu

As you can see in Figure 3-2, when first displayed, the main menu gives you a choice between six major alternatives:

- Set filters for capturing frames.
- Set a trigger to freeze the Capture Buffer during data collection.
- Start capturing frames.
- Display the current contents of the Capture Buffer.
- Use files to save or load data or setups.
- Exit to DOS.

## Preparing to Capture

If you're about to collect data, you should move first to *Capture Filters* (to describe the data you want collected), then to *Triggers* (to describe how stop capturing and freeze the capture buffer), and finally to *Capture* (to describe the real-time displays you want during capture, and start capturing frames). Chapter 4 is devoted to the details of these choices.

## Preparing to Display

If you've just captured some frames, you may want to move first to the *Files* option to save your captured frames to a file. If you want to display frames, but as yet have none in the Capture Buffer, you may need the *Files* option to load the Capture Buffer with a file of previously-saved frames. Then you're ready for the *Display* option, to examine the frames in the buffer. Chapter 5 is devoted to the details of these choices.

## To Conclude Work

When you select Exit to DOS, work with the Sniffer terminates. But before you do that, if you've established filters, triggers, or display options, you may want to save a record of them in a file. Then at a future session you recall that file, and re-establish all the options as they were. To do that, before you exit, first select the *Files* option, and from there the option *Save Setups*. (See Chapter 5.)

## Conventions in the Sniffer Menus

The following conventions apply equally to the Sniffer's main menu and to the various subordinate menus you may reach from it.

- **Enter key.** At certain nodes in the menu tree, you need to press *Enter*, either to start the desired action, or to bring up a subordinate menu where you'll supply further details. Wherever *Enter* is a possible response, the node is marked with the symbol ↵ (to remind you of the "return" symbol engraved on the *Enter* key).
- **Check marks.** At certain nodes in the menu tree, you may select any number of alternatives from a list. Each of the items selected is checked, while each item not selected is marked with an x. For a highlighted item, you reverse the current choice by pressing the space bar. When you wish to reverse the setting of many items at once, hold down the Alt key while you press the space bar. That reverses the setting of all items at and below the cursor.
- **Radio controls.** At certain nodes in the menu, you may select one from a list of mutually exclusive alternatives. (They're called radio controls by analogy with the push buttons on a radio, which don't let you tune in two stations at once.) A set of mutually exclusive choices is shown linked by a vertical bar, with the symbol ▶ at the one selected. For a highlighted item, you select it (and deselect the one previously selected) by pressing the space bar.

A menu item which has *none* of these markings serves as a heading for a branch of the menu tree. You select it as route to one of the items subordinate to it, but you can't take action until you reach a specific item beneath it. For example, in the Main Menu (Figure 3-2) when you select *Files* you have to move from there to one of its subordinate menus before you can start action.

## Function Keys

The function keys **F1** through **F10** provide a quick and convenient way to indicate the action you want next. For each menu or display, the current meanings of the function keys are shown in a band across the bottom of the screen. A button is shown for each key which is active at the moment, with a label to describe its action. (Keys which are not active simply disappear from the display.)

## **Help**

In almost every situation, *F1* calls up a list of help topics. You may scroll to the topic you want, and then press *Enter* to see a brief discussion of that topic. (The only exception is that you can't signal for help while the Sniffer is actively capturing data. You have to press *Pause* to temporarily suspend data capture, or wait until you've finished capturing.)

# Chapter 4. Capturing Frames

In order to capture frames from the network, you have to do the following:

- If the Sniffer isn't already connected to the network, plug its token-ring connector to an available access unit.
- If you wish, set capture filters to tell the Sniffer which frames to discard and which to capture. (You can do that explicitly, or by loading a previously-saved setup file.)
- If you wish, set a trigger to tell the Sniffer how you want it to stop capturing frames and freeze the Capture Buffer. (That too can be done explicitly or by loading a previously-saved setup file.)
- When you're ready to start capture, press **F10**, or move to *Capture* in the main menu and there press *Enter*..

When you're collecting data to look at a specific problem, you'll probably want to set up a capture filter and a trigger before you start capturing. But when you're just browsing to check the network's traffic load, you can start right away without either.

## Inserting the Sniffer into the Ring

When you plug the token ring connector cable into an access unit on the ring, that makes a physical connection possible, but does *not* yet insert the Sniffer electrically into the ring. (The access unit is designed so that it does not in fact connect a computer until the computer sends an explicit signal to do so.) Insertion is done automatically, as necessary, when you tell the Sniffer to start capturing frames. Insertion takes a few seconds. You'll see a message window headed "CAPTURE" which at first says "Resetting the Adapter" and then "Inserting onto the Ring."

After you stop capture, the Sniffer remains inserted. Unless you do something that causes it to disconnect, it will be able to start immediately when you resume capturing frames. You won't again experience the delay or see the message about insertion.

Loading a saved file into the Capture Buffer causes the Sniffer to disconnect itself from the ring, even though its cable remains physically connected. Thus, if you look at some saved data and then return to capturing new data, the Sniffer will again have to insert itself in the ring, and you will again see the message that it is doing so.

## Setting the Capture Filters

From the main menu, move up to select *Capture Filters*, and from there to the right to select one of the three types of filter: station address, protocol, or pattern match.

When you set several filters, the frames that get through are those that meet all the criteria: with the station address *and* the protocol *and* the pattern you specify.

### Station Address Filters

Before you set an address filter, by default the Sniffer accepts a frame from any station to any station (Figure 4-1). To change that, use the arrow keys to highlight the line labeled *From* or the line labeled *To*.

The Sniffer will accept a frame when:

- Its addressee is the *To* address you named  
*and*
- its source is the *From* address you named  
(or also the reverse, if you checked *reverse direction*)  
*and*
- It's a member of the broadcast/non-broadcast categories you've checked.

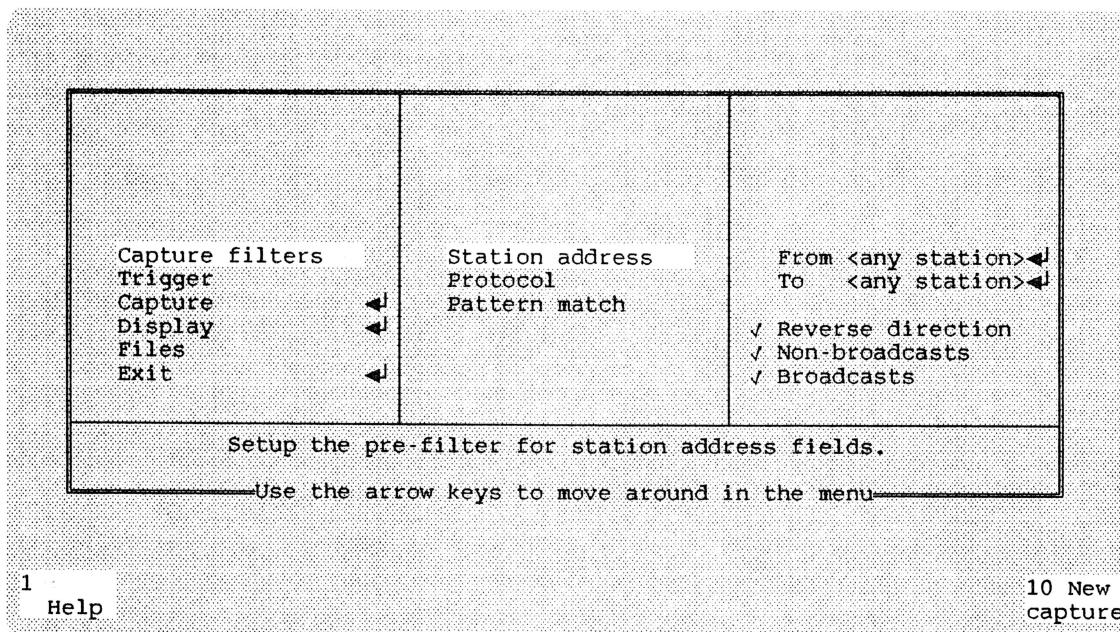


Figure 4-1: Default settings of Capture Filters for station address.

When you press *Enter* on one of those lines, the Sniffer opens a window in which you can see the list of names and station addresses in your current names file (Figure 4-2). If the name you want is appears in the list, scroll to it and press *Enter*. The Sniffer closes the window, and inserts the name you've selected in the display.

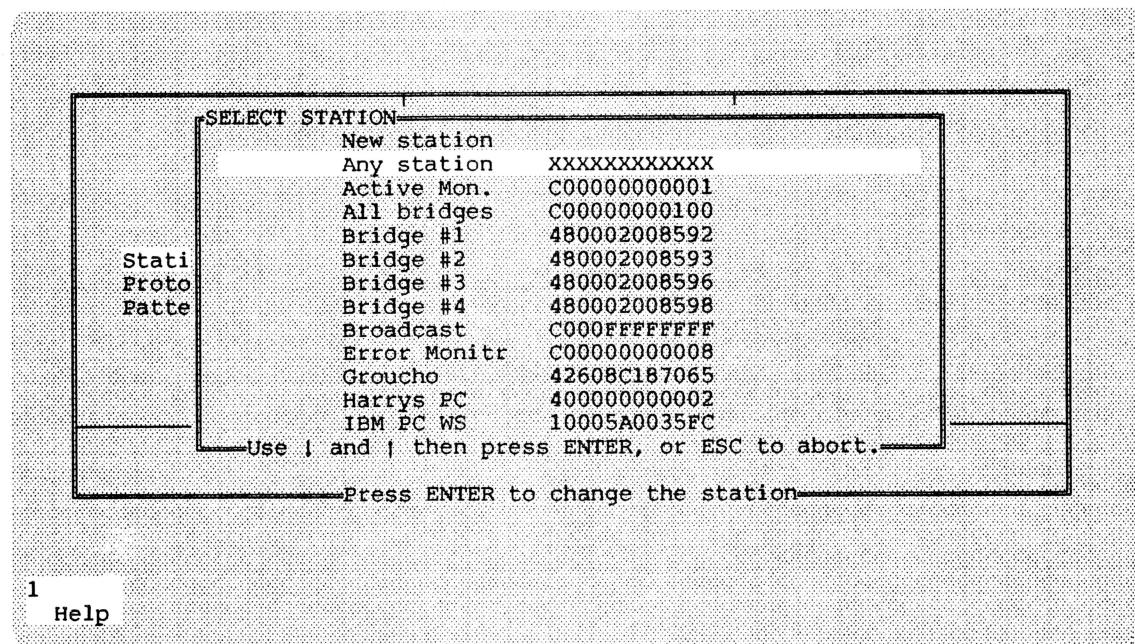


Figure 4-2: Menu to select a station for a station address filter.

When you want the Sniffer to accept frames from any source, select the line that says *Any station*. This entry isn't really part of the names file, but the Sniffer always shows it at the head of the list of stations. It always has `xxxxxxxxxxxx` for its address field. When you select *Any station*, the Sniffers won't test that field, so any frame will pass.

When the station you want to select isn't on the list, select the line marked *New Station* and press *Enter*. The Sniffer opens a window in which you can enter a symbolic name and its hexadecimal station address (Figure 4-3).

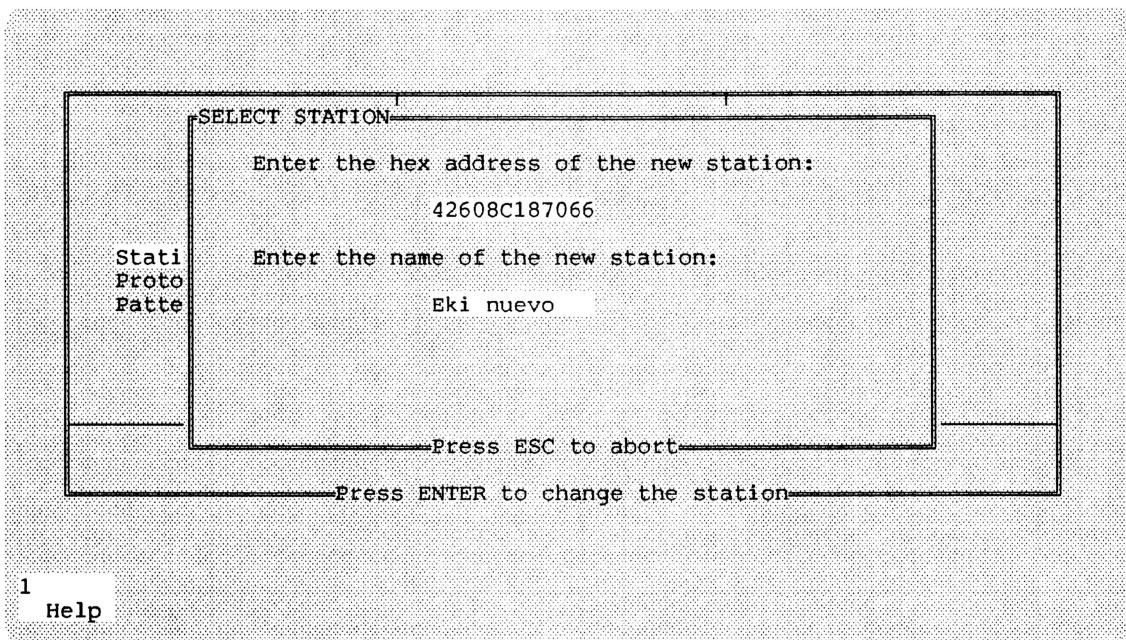


Figure 4-3: Window for inserting a new name and station address.

**Watch out:** Inserting a new name in this manner revises the namelist in working memory. It doesn't update the stored file of names, so the revision is temporary. If you want to make it permanent, select *Display* in the main menu, then *Managing Names*, and finally *Save names*.

Within the Sniffer's name list, there can be only one name for each station address. When you specify a name for a station that already had one, your new name thereby replaces the former name. (You can edit the list of names from the *Edit names* submenu, described in Chapter 5.)

A station may direct a transmission not to another specific station but to a *functional* address. That may be a role played by one or another station, or by no station, or it may serve as a collective name for a group of several stations. Such addresses have the high-order bit on (and therefore are shown in hexadecimal by a number starting with 8 or more), whereas no individual station has an address with that bit on. You may assign a name to a functional address (for example "Error Monitor" or "LAN Manager"). Of course, that name will turn up only as an addressee, never as a sender of frames.

## Protocols in the Capture Filter

In the main menu, move to *Capture Filters* and then *Protocol*. You'll see in the right panel the list of service access points (SAPs). A check mark indicates a SAP that will be accepted. By default, all are initially checked (Figure 4-4).

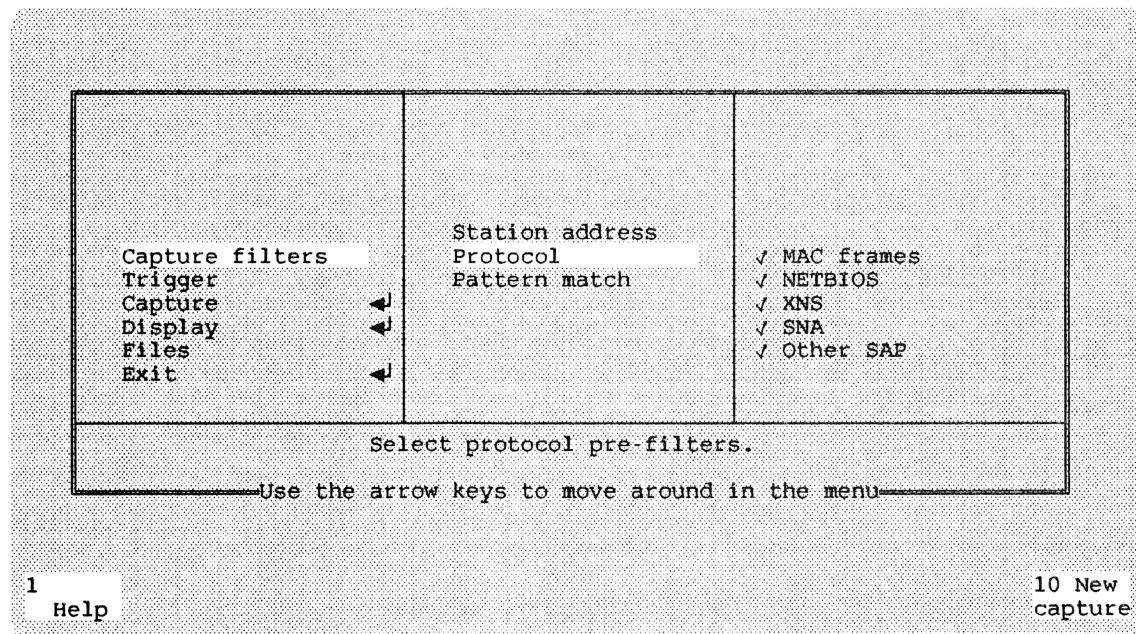


Figure 4-4: Menu to select SAPs for the Capture Filter.

Using the up or down arrows, move to each entry you want to change. Press the space bar to reverse its status.

During collection, the Sniffer accepts any frame whose SAP is checked on this list.

**Watch out:** Specifying a protocol in this manner revises the capture filter in working memory. It doesn't update the stored file containing your Sniffer setup, so the revision is temporary. If you want to make it permanent, when you have completed this and other changes to your setup, move to *Files* in the main menu, then *Save*, and finally *Setup*.

## Pattern Matching in the Capture Filter

To capture only frames containing a particular pattern, move to *Capture Filters* and then *Pattern match*. In the right panel you'll see a display of the current pattern and offset. By default, these are initially xxxx ("don't care") and the position is given as frame-relative offset 0 (Figure 4-5).

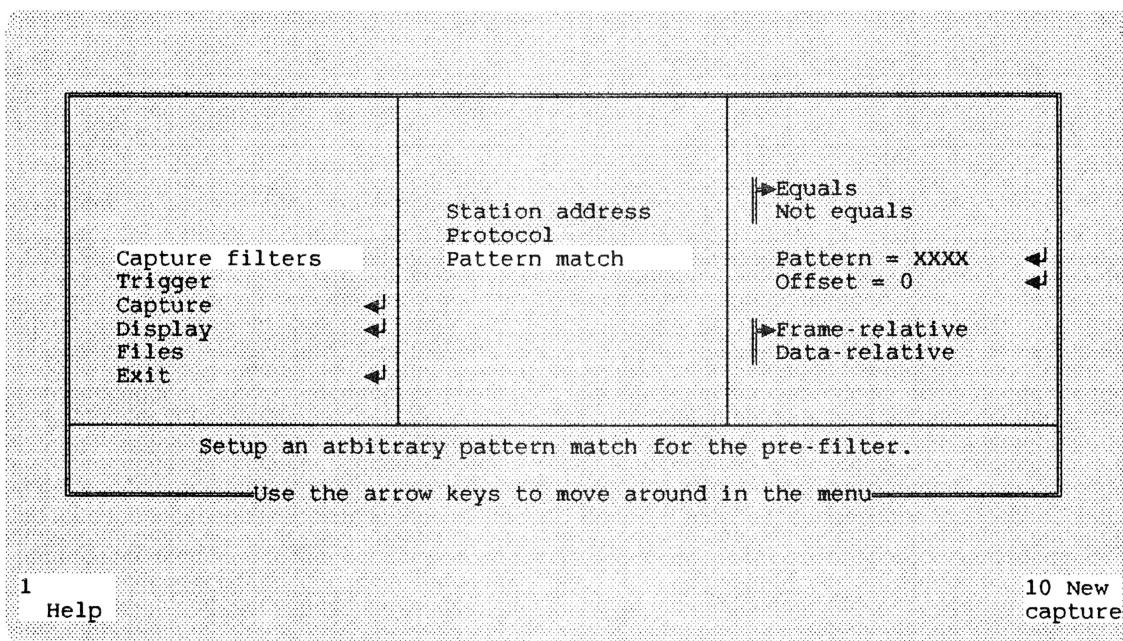


Figure 4-5: Specifying pattern match for the Capture Filter.

The pattern match filter may be set to pass only frames that have the specified pattern, or only those that do not. Move to the line *Equals* or *Not equals* and press the space bar to select the one that's highlighted (and thereby deselect the other).

To specify the pattern to be tested, move the cursor to highlight *Pattern* and press *Enter*. The Sniffer opens a window that allows you to specify a new pattern. A pattern may occupy from one to four half-bytes (nibbles). Insert the pattern you want where the current pattern is displayed. You must compose your entry from the hexadecimal digits 0 to F, or the code x, which means "don't check this nibble."

To identify a pattern, you must also describe its position. Its position is measured as its offset to the beginning of the four-nibble pattern, stated in bytes. The offset may be measured either from the beginning of the frame, or from the beginning of the frame's data field (the first byte after the source address and the variable-length routing-information field). These are called "frame-relative" and "data-relative" offsets, respectively. To select one (and thereby deselect the other), move the cursor so that *Frame-relative* or *Data-relative* is highlighted, and press the space bar.

To insert an offset, move the cursor to highlight *Offset* and press *Enter*. The Sniffer opens a window (like the one for pattern shown later, in Figure 4-9). You may insert the decimal value of the offset you want. Figure 4-6 illustrates the second part of such a change. (In the example, the pattern being matched is the *duplicate address test* in a MAC frame, emitted by a station inserting itself into the ring. Such a frame has '07' at offset 17.)

**Watch out:** You must express the size of the offset to the start of the pattern as a *decimal* number of bytes. Don't mistakenly substitute a hexadecimal address. Note that the hex view not only displays the entire frame in hexadecimal, but also uses hexadecimal labels for the rows of its display. If you're looking up a pattern in the hex view, remember to translate its offset to decimal before you type it as the offset of your pattern.

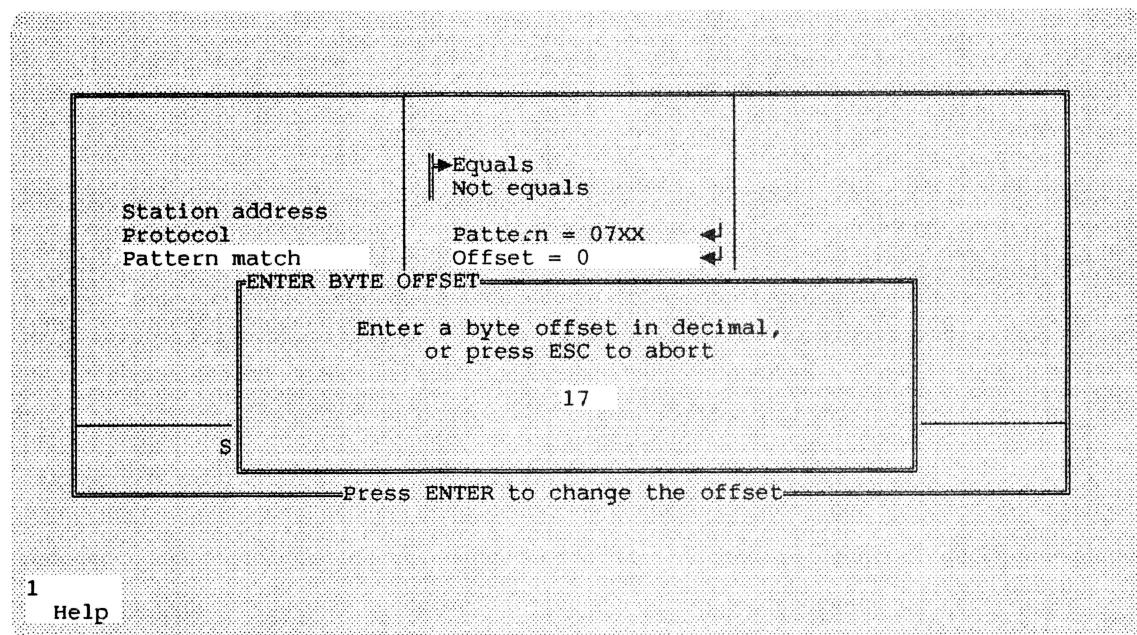


Figure 4-6: Inserting a pattern and offset for the Capture Filter. The pattern 07xx has already been supplied; the window is ready to receive the offset.

**Watch out:** Specifying a pattern in this manner revises the capture filter in working memory. It doesn't update the stored file containing your Sniffer setup, so the revision is temporary. If you want to make it permanent, when you have completed this and other changes in your setup, select *Files* in the main menu, then *Save*, and finally *Setup*.

## Setting the Trigger

The trigger pattern consists of one to four half-bytes (nibbles) at a specific position in a frame. The pattern and its location are defined in the same way as the pattern used in the capture filter; you set them in a similar window. In the main menu, select *Trigger*. Then move right to the list of trigger options (Figure 4-7). The pattern xxxx indicates no trigger.

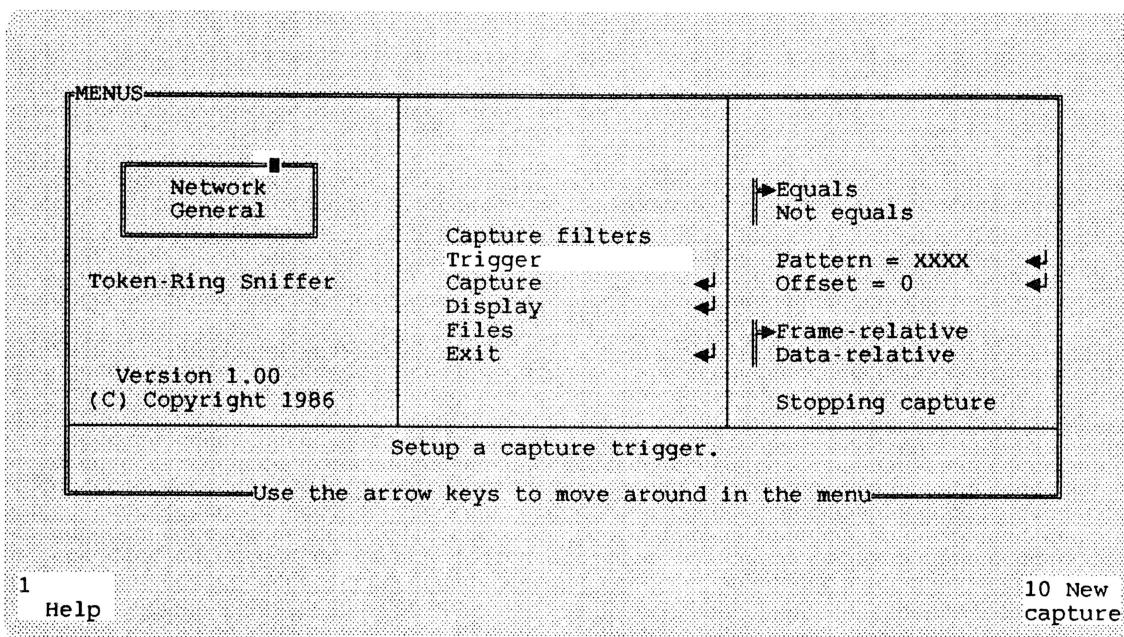


Figure 4-7: Default settings of the Trigger.

To set a trigger pattern, use the up or down arrows to highlight *Trigger*, then move right to select a line in the list of trigger options. These allow entries for *Pattern*, *Offset*, *Equal/Not Equal*, and *Stopping Capture*.

For example, to trigger on a frame containing the SMB error return code, you might first set the Capture Filter to pass only NETBIOS frames, and then set the Trigger to signal detection of a value other than the normal return code 00. In an SMB frame, the return code is at data-relative offset 39.

When you move the cursor to highlight *Offset*, the Sniffer opens a window in which you may write the desired offset (Figure 4-8).

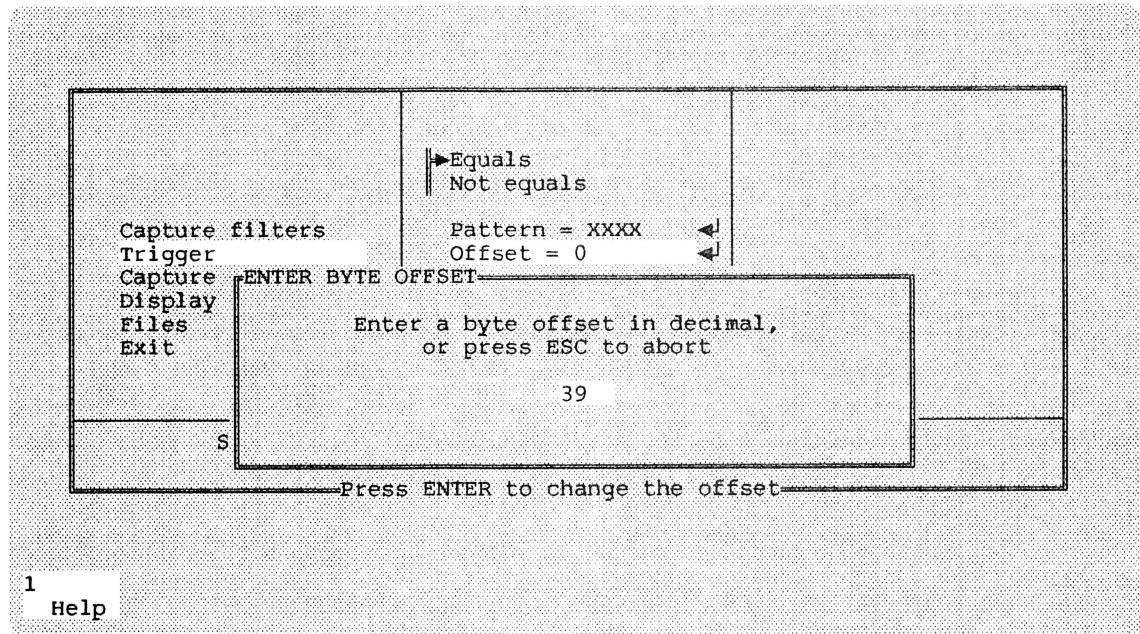


Figure 4-8: Window to supply offset while specifying a trigger pattern. The pattern requires both the pattern text and the offset; in this example, the offset is being entered before the pattern has been specified.

Since the event you want to detect is a frame which does *not* have 00 at that location, use the arrows to highlight *Not equals*, and press the space bar to select it (and deselect *Equals*).

To fill in the trigger pattern, highlight *Pattern* and press *Enter*. The Sniffer opens a window in which you can enter the appropriate pattern (Figure 4-9).

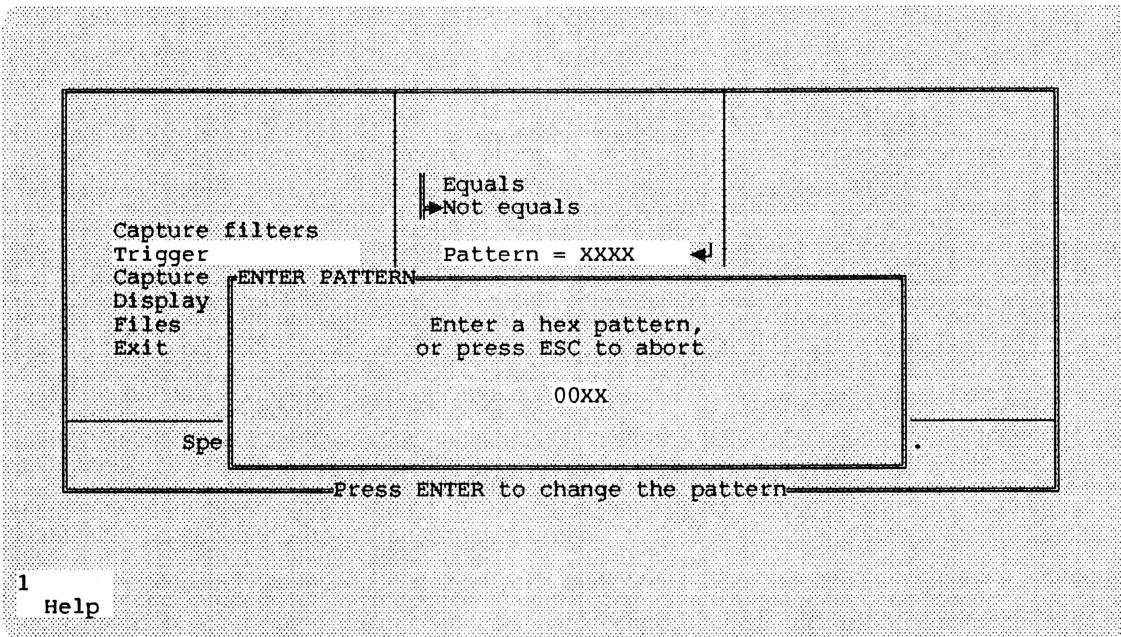


Figure 4-9: Window in which to supply Trigger pattern.

The window has space for two bytes. Since the test involves only one, leave xx in the other. The code xx means that the byte is not checked (so the *Equal* or *Not equal* setting doesn't apply to it.)

As with patterns used in the frame capture filter, you must indicate whether the offset is from the beginning of the frame, or from the beginning of the frame's data field. These are called "frame-relative" and "data-relative" offsets, respectively. To select one (and thereby deselect the other), move the cursor so that *Frame-relative* or *Data-relative* is highlighted, and press the space bar (Figure 4-10).

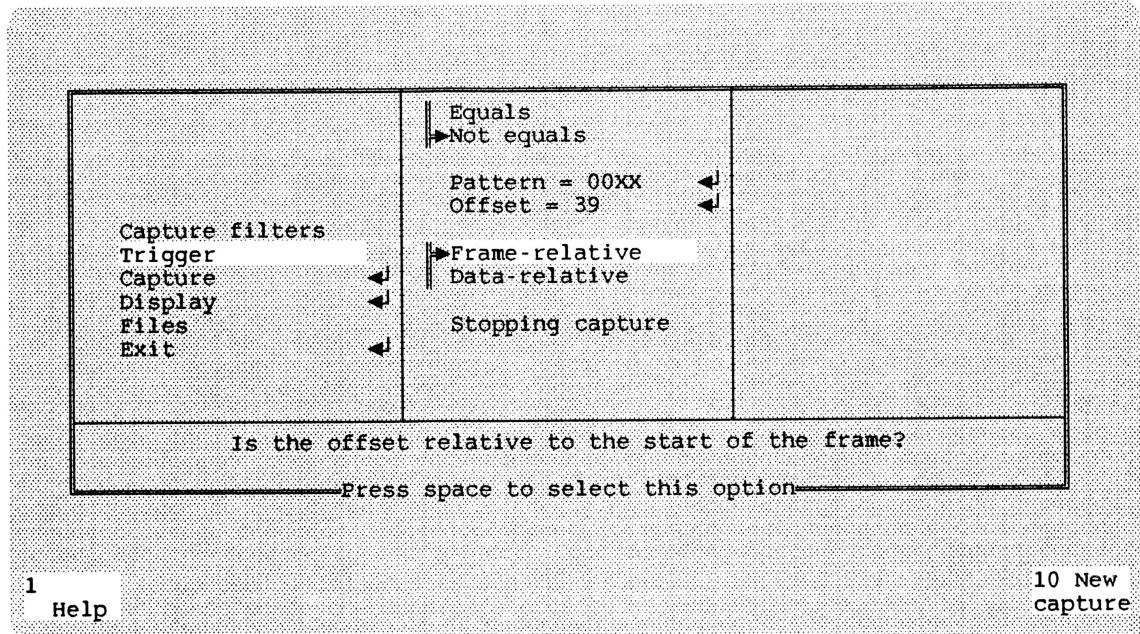


Figure 4-10: Specifying whether the trigger pattern's offset is frame- or data-relative.

### When to Stop Capture: Positioning the Trigger Frame in the Capture Buffer

To complete description of a trigger, indicate when you want the Sniffer to stop capturing data. If the Sniffer stops capturing as soon as it sees a matching frame, the trigger frame will be the *last* frame to arrive. The other frames in the buffer will be those that *preceded* the trigger.

Conversely, if the Sniffer doesn't stop until the trigger frame is about to be pushed out the far side of the Capture Buffer, the other frames in the buffer will be those that *followed* the trigger frame.

To indicate your choice, move the cursor to highlight the one you want, and press space bar to select that one (and deselect the others). Figure 4-11 shows a selection which will stop capture when 75% of the buffer space is used by frames that preceded the trigger, and the remainder by frames that followed it. (Note that the value you specify determines the proportion of buffer *space* before or after the trigger frame, not the number of frames.)

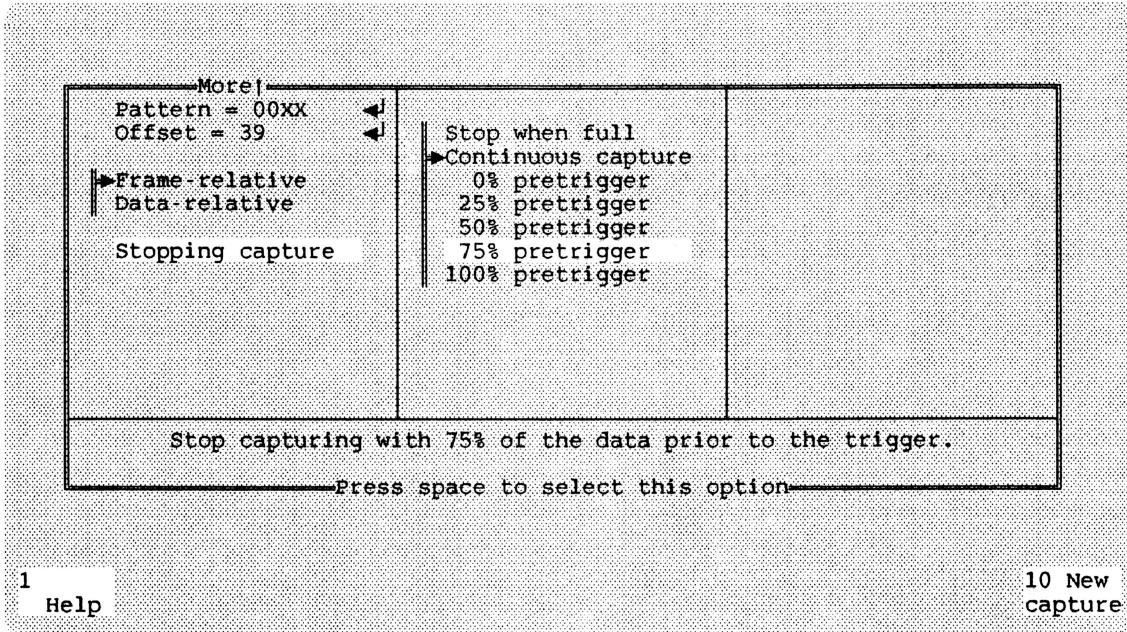


Figure 4-11: Selecting the rule for stopping capture.

## Marking the Trigger Frame

When the Sniffer matches the trigger pattern, it reports that fact by changing the label on the screen (Figure 4-13) from CAPTURING to TRIGGERED. In the display, the trigger frame is marked with the letter T. (Later, during display, you can jump to the trigger frame, and perhaps mark it so as to display time relative to it.)

The Capture Buffer never contains more than one frame marked T. If your reply to *Stopping Capture* permits capture to continue after a trigger frame has been spotted, and the Sniffer subsequently admits other frames that match the trigger pattern, only the first is considered the trigger frame and only the first is reported and marked. However, if you press *Pause* when a trigger frame is in the buffer, and then if you later press *Resume* without clearing the buffer, the arrival of a later trigger frame will be reported; it will be marked with a T and the T will be removed from the earlier trigger frame.

## **Stopping When the Buffer is Full**

This option is most often used when you haven't set a trigger. Nevertheless, it still has effect if you combine it with a trigger pattern.

When you select *Stop when full* but the Capture Buffer fills before the Sniffer finds a trigger frame, capture nevertheless ceases. There will be no frame marked T in the buffer.

When you select *Stop when full* and the Sniffer finds a trigger frame before the Capture Buffer is full, it continues capturing frames until it has filled the buffer. The trigger frame will be marked with a T as usual, but its position in the buffer will depend solely on how full the buffer was when the trigger-frame arrived.

## **Continuous Capture**

This setting instructs the Sniffer not to stop capturing frames until you signal manually from the keyboard. If a trigger frame happens to arrive, the first such frame will as usual be reported and marked with a T. The trigger frame will enter the Capture Buffer. But since capture doesn't stop, other frames arriving after it may push it out of the buffer before you call a halt.

## **Capturing Frames and Real-time Display**

Once you've set up appropriate filters and a trigger, you're ready to capture frames. As capture proceeds, the Sniffer displays a running total of the frames it has seen, and a sorted count of those that the Capture Filter has accepted. A dynamically repainted bar graph shows the rate of data transmission at each moment, and the "high water" maximum during the session.

Before you press the button to start capture, you may select from several options regarding the screen that the Sniffer will display during capture. (Like the particulars of the filters and the trigger, the state of these display options is included as part a saved setup, so you can restore all these options at once by loading a saved setup file; see the discussion of *Setup Files* in Chapter 5.)

To prepare to capture frames, select *Capture* in the main menu (Figure 4-12). The Sniffer won't actually start capturing frames until you press **F10** (labeled *New Capture*), or move to *Capture* and while it is highlighted, press *Enter*.

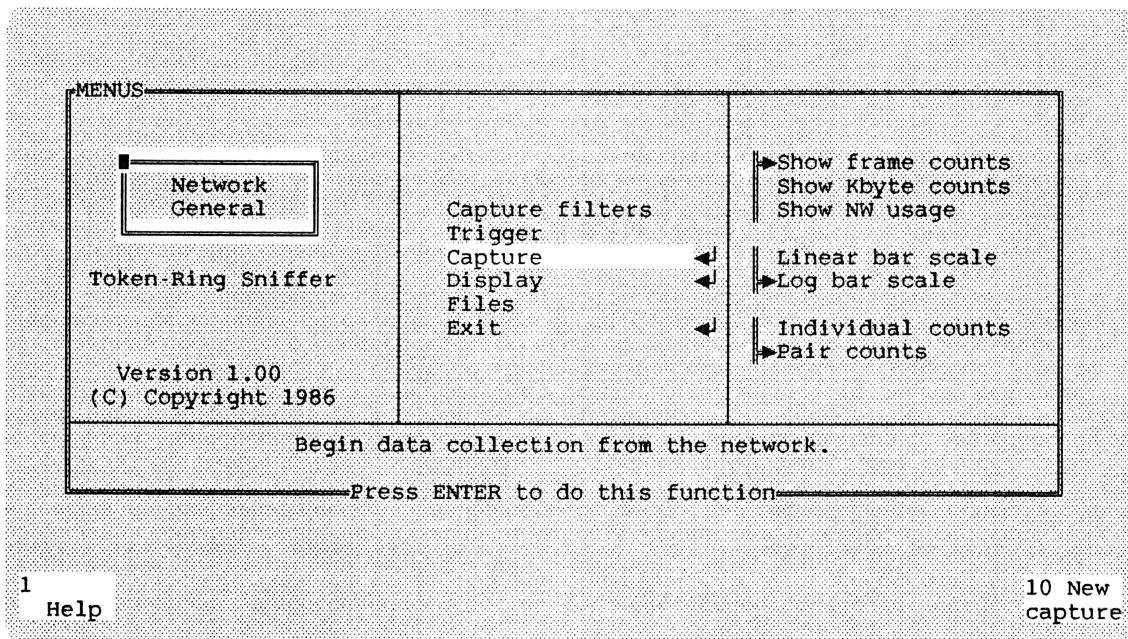


Figure 4-12: Capture option in the main menu.

Before you press *Enter* or **F10**, check the panel at the right. It specifies the type of display you'll see during capture. If it's satisfactory, start capture by pressing *Enter* or **F10**. But if you need to revise the settings of the capture display, first use the arrow key to move to the panel at the right, and there set the options you want.

## Totals

In the top right corner of the capture display, the Sniffer records the total time (in hours, minutes and seconds) during which the counts on the screen were accumulated. The elapsed time counter doesn't advance during a pause, and is reset to zero whenever you clear the counters.

On the row separating the counters from the traffic-density bar graph, the Sniffer displays three grand totals:

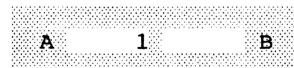
- The total number of frames seen (including both those accepted and those rejected by the Capture Filters).
- The total number of kilobytes accepted since capture began (including both frames still in the Capture Buffer and those discarded to make room for later arrivals).
- The total number of frames accepted since capture began (including both frames still in the Capture Buffer and those discarded to make room for later arrivals).

## Traffic Counts

For all the frames accepted by the Capture Filters, the Sniffer tabulates a set of running totals. Each incoming frame is tabulated either by the station that sent it, or pairwise by sender-and-addressee. The tables are updated in real time, as frames arrive. Before capture starts, the table is empty. As soon as a frame arrives, the Sniffer notes which station sent it (and, when you're tabulating pairwise, to whom it's addressed). When a frame with that sender (or sender-addressee pair) has been seen before, the Sniffer updates the count for the corresponding entry in the table. But when the frame represents a station (or station pair) that hasn't previously been heard from, the Sniffer creates a new entry in the table. The Sniffer builds the table as frames arrive. Within the screen, a particular entry's position depends on the arrival sequence. Positions in the table are filled from left to right, then top to bottom.

## Pairwise Tabulation

When you elect pairwise reporting of traffic, the Sniffer allocates half a line for each pair (Figure 4-13). If the first frame received was sent from station A and was addressed to station B, the Sniffer creates an entry for the A/B pair, like this:



The number 1 indicates that 1 has been sent from A to B. At the same time, a space is reserved to count traffic from B to A.

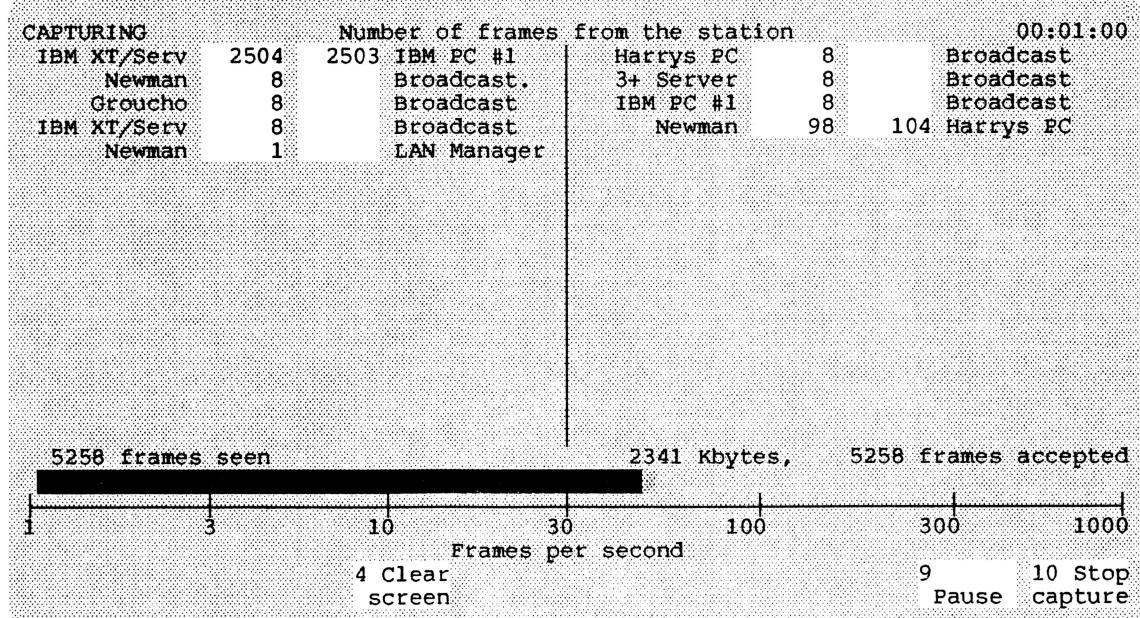


Figure 4-13: Pairwise tabulation during capture, by sending station and addressee.

Although traffic can only originate from a physical station, it may be addressed to a group (for example, *Broadcast*) or to a role (for example, *Error Monitor*). When such names appear in the addressee field of a pairwise tabulation, you'll see traffic addressed to those names but never traffic from them.

This table has 17 rows, and hence a total capacity of 34 pairings of sender and addressee. If the incoming frames have more distinct pairs than that, they go to the Capture Buffer as usual but they're not included in the screen tabulation.

## Tabulation by Sender

When you elect *Individual counts*, traffic is tabulated solely according to the station that sent it (Figure 4-14). The Sniffer divides the screen into four columns, and adds entries from left to right as it notices new senders. This table has 17 rows, and hence a total capacity of 68 stations. If the incoming frames have more distinct senders than that, they go to the Capture Buffer as usual but they're not included in the tabulation.

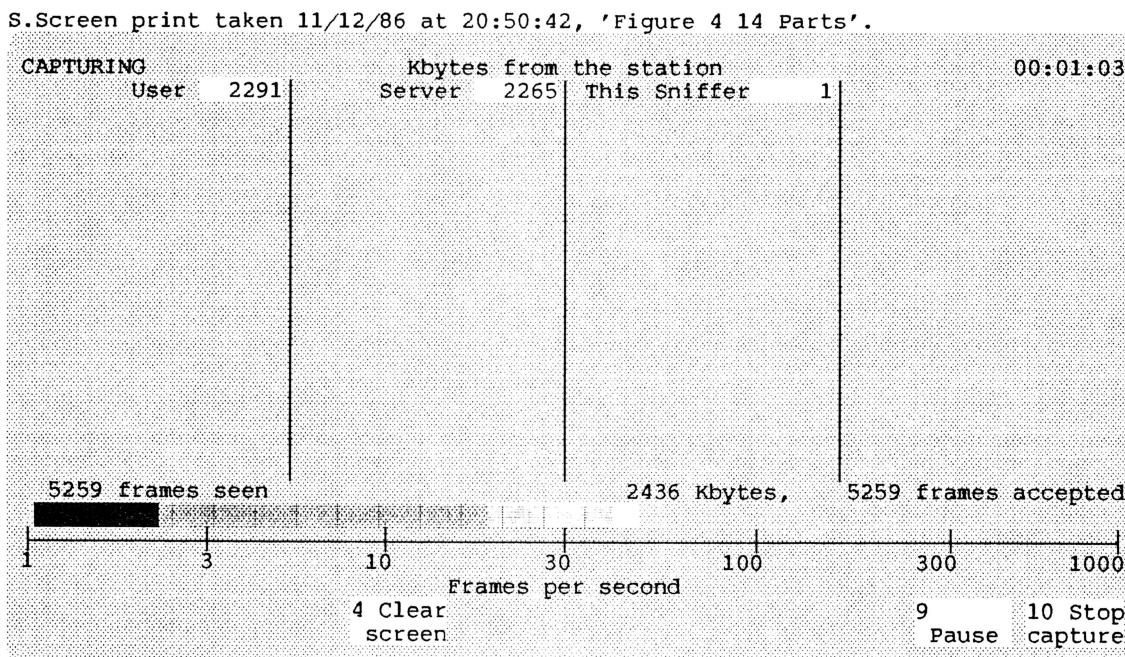


Figure 4-14: Individual tabulation by sending station during capture.

## **Counting Frames, Kilobytes, or Percentage Utilization**

You have a choice of three ways of reporting traffic. These choices affect both the tabular counts and the dynamic bar graph. The options are:

- **Frames.** The counters show actual numbers of frames, up to 999,999. The bar graph shows frames per second, with a maximum of 1,000 frames per second.
- **Kilobytes.** The counters show kilobytes transmitted, with fractional amounts rounded up to the next higher integer multiple of 1024 bytes. The bar graph shows kilobytes per second, with maximum of 500 kb/sec.
- **Percentage network utilization.** The counters show kilobytes as described above, but the bar graph show percentage of total network bandwidth (500,000 bytes/sec) from 0 to 100%.

## **Real-time Traffic Density Bar Graph**

During capture, the display shows a continuously updated horizontal bargraph showing network utilization during the last second. The graph shows the current reading as a solid bar, and the maximum reached since capture started as a dotted bar.

The bar graph uses the units selected (frames per second, kilobytes per second, or percentage utilization), on either a linear or a logarithmic scale. The advantage of the logarithmic scale is that a displacement of one unit represents the same proportional change in traffic density regardless of where on the scale it occurs, and that small values are more visible.

## **Unrecognized Addresses Noted During Capture**

When you first start the Sniffer, it sets up a definition table of station addresses and their symbolic names. (Initially, it builds the table by reading from the file STARTUP.TRD.) While it's recording entries in the counter table, the Sniffer looks up each address in its definition table, and substitutes the symbolic name in the table it displays.

For an address retained in the Capture Buffer but not represented in its table of definitions, the Sniffer does two things:

- Represents that station in the counter display by its hexadecimal address rather than a symbolic name.
- Adds the new address to the definition table, but with blanks in the field for its symbolic name.

After you've completed capture, you can go the Files menu, select Manage names, and there edit the entries in the definitions table.

You'll see there at the head of the list all the addresses that the Sniffer has encountered but for which it had no symbolic names. You can edit those entries to provide symbolic names. Then, when you go to display data from the Capture Buffer, the display routines will augment the station addresses with the symbolic names you've provided. (Note that the definitions table thus edited is the working copy. It won't be retained next time you start the Sniffer unless explicitly request to *Save names*, and thereby rewrite the file STARTUP.TRD.) See the discussion of *Managing names* in Chapter 5.

## Options During Capture

During capture, the following function keys are active:

- F4** *Clear screen.* This removes the counter table from the screen, and resets the counter clock to 00:00:00. The process of counting starts over, building a new counter table. However, the Capture Buffer is unaffected, and new station names added to the names table remain there.
- F9** *Pause.* This suspends capture, so that arriving frames are not seen, but leaves the Capture Buffer and the counter tables unchanged. From the paused state (see below) you can if you wish resume capture, perhaps after changing the way traffic is counted (which resets the counters but doesn't change what's in the Capture Buffer).
- F10** *Stop Capture.* After capture has stopped, you may save the contents of the Capture Buffer to a file, or display what's there. If you subsequently start a new capture, the present contents of the Capture Buffer will be discarded (but the Sniffer asks you to verify that you mean to discard them, and lets you withdraw the request).

## Options During Pause

When you press **F9** (*Pause*) during capture, you may choose from among several function keys visible on the screen. Some of them permit you to add more frames to those already captured, and some do not.

- F3** *Data display.* Terminates capture, discards the counters, and starts display of the frames in the Capture Buffer.
- F4** *Clear screen.* Resets the timer and discards the counter tables but does not discard data from the Capture Buffer (same as **F4** during capture, above).

- F5** *Menus.* Terminates capture, discards the counters, and shows you the main menu. Frames in the Capture Buffer are unaffected. From the main menu, you may elect to save the Capture Buffer in a file, edit and save the revised name table, start display of the Capture Buffer... or elect any of its other options.
- F6** *Capture Options.* Takes you to the screen which permits you to choose options for the screen tabulation of incoming data. If you there select a different option, resets the counters (like F4, above). You may then if you wish resume capture.
- F9** *Resume.* Resumes the examination of incoming frames, accumulating those accepted to the Capture Buffer, and adding to the counters (which may or may not have been reset during the pause).
- F10** *New Capture.* Discards the Capture Buffer and the counters (but leaves the working copy of the name table unaffected), and starts afresh with the process of capture.

If you have not saved the contents of the Capture Buffer, the Sniffer shows you a screen asking whether it is OK to discard the frames now accumulated there. If you assent, it goes ahead, discarding the Capture Buffer and starting the process of capture afresh. If you want to save the frames in the Capture Buffer, you should press Esc to abort this command, then press F5 to return to the main menu, where you can select *Files* and then *Save*. After that, you can if you wish again press **F10** to start capture.



# Chapter 5. Displaying and Interpreting the Captured Data

This chapter describes the various ways you can display and analyze the network frames that you've captured. (The Sniffer also provides meters and counters that summarize the rates at which data are arriving; they're described in Chapter 3.)

To manage *how* frames are viewed, or select *which* frames to look at, you start from the *Display* branch of the main menu (see Figure 5-1).

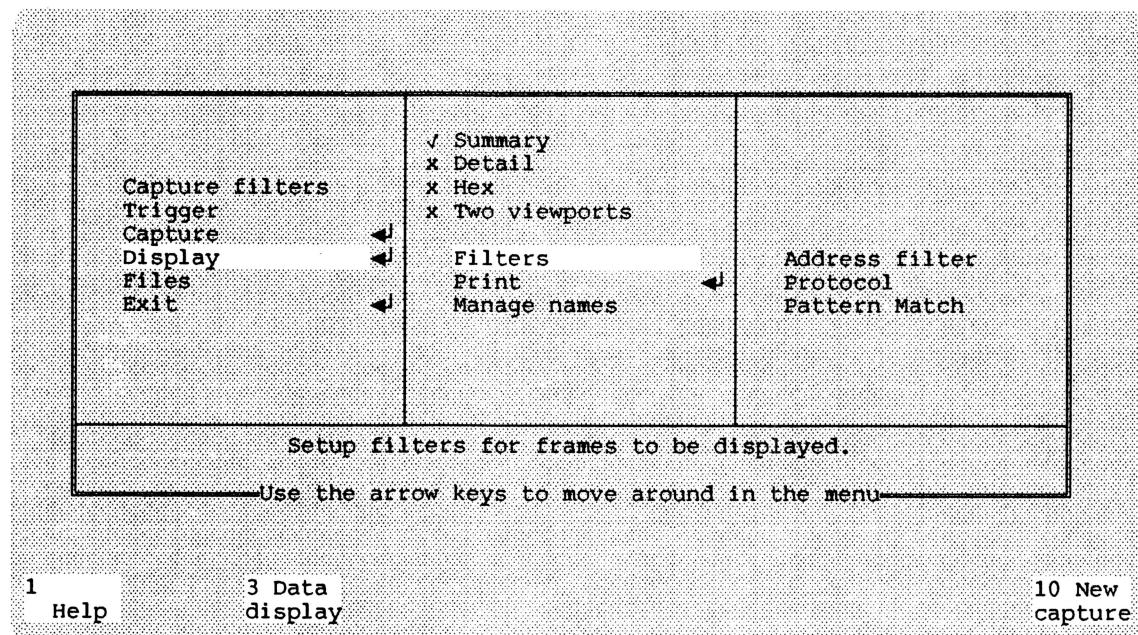


Figure 5-1: The main menu, showing the *Display* option and its principal branches.

However, to *save* the current contents of the Capture Buffer to a file, or to *load* a saved file into the Capture Buffer, you start from the *Files* menu (whose entry point is visible at the left in Figure 5-1, just below *Display*).

## Deciding Which Set of Captured Data to Display

Before you can display or analyze them, frames must be in the Capture Buffer. You can display frames that have just arrived (and may not have been saved). Alternatively, you can load into the Capture Buffer a file that you saved earlier, representing the frames that were in the Capture Buffer at an earlier time.

When you load a saved file, the file you load replaces whatever was then in the Capture Buffer. (So if you want a record of the Capture Buffer, you must save it before you load something else.)

While you're displaying what's now in the Buffer, the Sniffer can't write new data there. Displaying therefore suspends the capture of data from the network.

## Loading a File of Previously-Saved Frames

To bring a previously-saved file into the Capture Buffer, select the *Files* menu and then *Load*. This option may apply to data or to setup files, but the default is data (Figure 5-2). Press *Enter*.

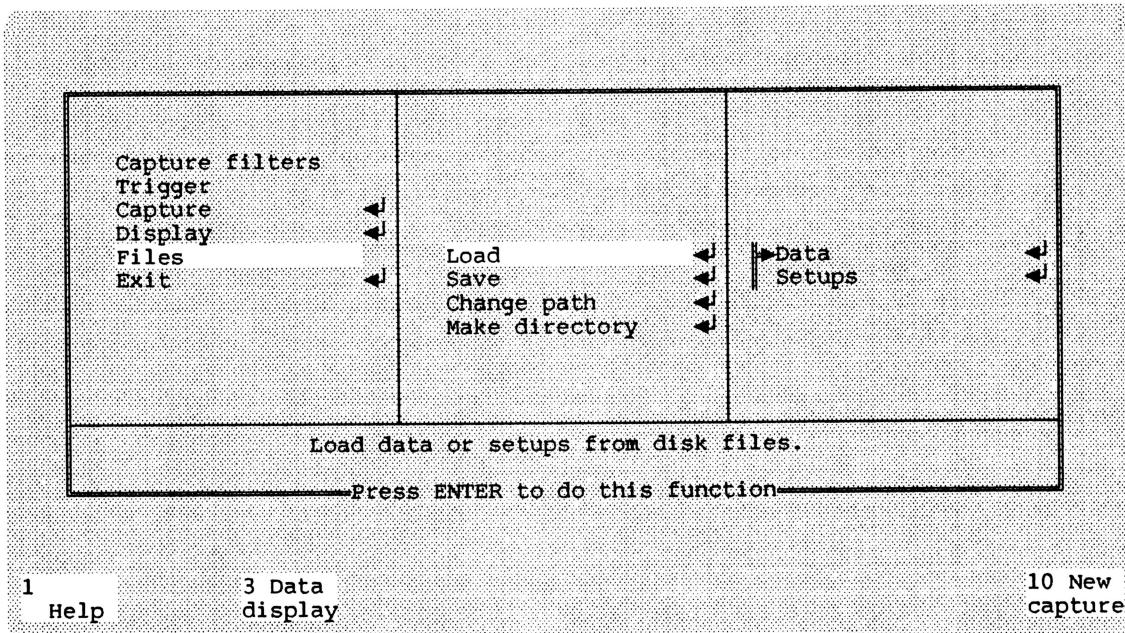
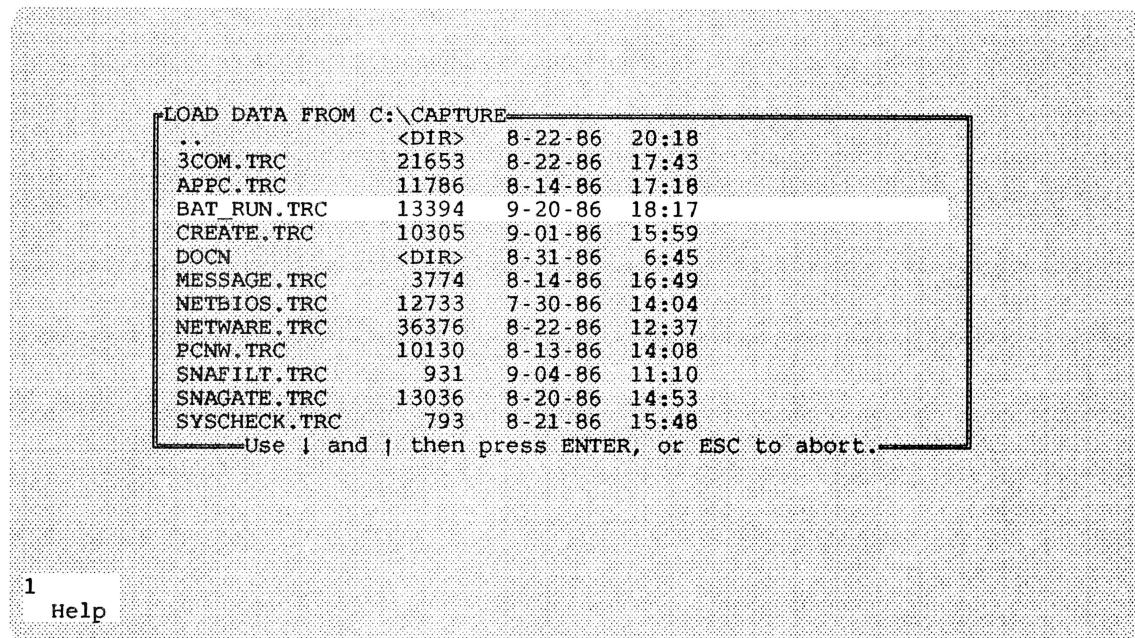


Figure 5-2: Main menu, showing choices you select to load the Capture Buffer with data from a file.

When you press return, the Sniffer shows you a list of capture files (Figure 5-3). Capture files are identified by the extension .TRC. The list is in alphabetical order. The name of the first file is highlighted. Use the up or down arrows to move the highlight to the file you want, and press *Enter* to load that file.

To jump directly to a part of the list, type a letter from the keyboard. The highlight moves directly to the next file starting with that letter.



*Figure 5-3: List of saved files that can be loaded to the Capture Buffer.*

The display lists all the capture files in the current directory (that is, the directory from which you started the Sniffer). When you keep TRSNIFF.EXE and the capture files in different directories, you'll probably find work more convenient if, before you start the Sniffer, you make your current directory the one that contains the files.

### **Switching to Another Directory from within the List of Files**

As you can see in Figure 5-3, some of the entries in the list are not files but directories. In that example, the name DOCN is a subdirectory of CAPTURE.

If you highlight a line that represents a directory and press Enter, the Sniffer changes the display to show a list of the files in that directory. The notation .. <DIR> in the top row indicates the directory one step nearer the root directory.

Selecting a directory line in the files display has the same effect as changing the path, described in the next paragraph.

## **Setting a Path to a Different Directory**

When the files you want to work with are in a directory different from the current directory (i.e., different from the directory from which you started the Sniffer), you can establish a path to the directory you want. You do that by the following steps:

- Return to the *Files* option
- Select *Change path*
- Type the complete path to the directory in which you want the Sniffer to look for files to load and to store files that you save.

## **Creating a New Directory**

When you're about to save some files, you may wish to create a new directory to contain them. You can do that without leaving the Sniffer. Within the *Files* option, select *Make Directory*. When prompted, type the name of the new directory.

DOS will interpret the name you write with respect to the current directory. For example, if your current directory is \CAPTURE and you ask for a directory called DATA, the Sniffer passes your request to DOS, and DOS creates a directory whose full name is \CAPTURE\DATA.

When you don't want the new directory to be a subdirectory of the current one, write its complete path: make the first character \ to show that the path you're writing starts from the root directory.

Creating a new directory does not of itself cause files to be saved into it. It just creates it. If you also want to use the new directory, then use *Change path* to indicate that.

## **Numbering of Frames in the Capture Buffer**

At the screen, you can scroll through the frames in the Capture Buffer, bringing one at a time to the screen (or, when you select the summary display, a frame and its neighbors).

Frames are identified by their sequence in the Capture Buffer. The first frame is referred to as Frame 1, the next as Frame 2, and so on.

You can *filter* the frames to be displayed so that you see only the frames that meet certain criteria. (See *Display Filters*, below.) Frames that don't meet your criteria are still in the Capture Buffer, but they're omitted from the display. Filtering out certain frames from the display doesn't change the numbering of those that remain visible. For example, when the filter excludes frames 2 and 3, in the display you see frame 1 followed by frame 4.

## **Options Apply Equally to Screen and to Printer Displays**

When you request a print-out of the Capture Buffer, your printed listing contains more or less the same information you'd see on the screen if you were to scroll all the way through all the visible frames in the Capture Buffer. The filters and display options apply both to the screen display and to the printed output. There are a few differences in the way things are arranged, and the *detail* view differs in its treatment of multiple levels; see *Printing a Report on Frames in the Capture Buffer*, later in this Chapter.

## **Display Filters: Selecting Which Frames to Display**

Either before you start the display, or as the display proceeds, you can establish a *display filter*. It selects a subset of the frames in the Capture Buffer, and limits the display to those.

When you subsequently save the contents of the Capture Buffer to a file, you have the choice of saving all the frames in it (regardless of the display filter), or just the frames that were accepted by the display filter then in effect.

## **Timing: When You Can Set Filters and Displays**

It's probably easiest to set up some sort of display filter and to select your initial form of display *before* you actually start displaying frames on the screen. However, it's easy to change filters or display options as you go along, after you've started the display. You do that as follows:

- From the display screen, Press **F6** to select *Display Options*.
- Revise your filters or display options.
- Press **F3** to return to the display.

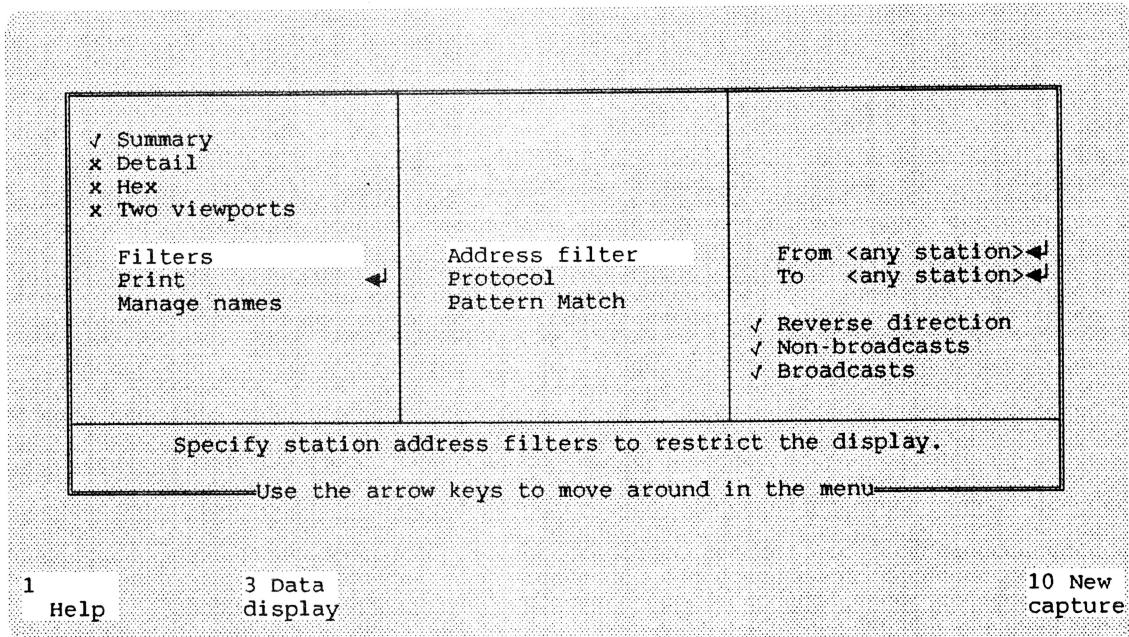


Figure 5-4: Menu to establish display filters.

The display filter selects frames according to three criteria:

- **Station address.** You can identify the source or destination by the symbolic address you've provided the Sniffer, or (if it has no symbolic address) by writing in its hexadecimal station address. The filter passes frames *from* the addresses you mention, *to* the addresses mention, or *between* the addresses you mention.
- **Protocol level.** Set a check mark beside the names of the protocols you wish to include. The filter passes each frame that contains *any* of the levels you mention.
- **Pattern.** You can specify a pattern of up to four half-bytes which must be present (or, optionally, absent) from the frames displayed.

## Procedure for Setting Display Filters

Address filters and pattern filters work in exactly the same way during display as they do during capture. However, during display, protocol filters are able to filter on embedded protocols, whereas during capture filtering is done by SAP, without consideration of higher-level protocols that may be embedded in a frame.

The procedure for setting the address filters, protocol level filters, and pattern filters during display are the same as those for setting capture filters, described in Chapter 4. You should turn to that chapter for details.

The frames eligible to appear in the display are those that meet any of the address criteria you specify *and* involve any of the protocol levels you identify *and* satisfy the pattern you specify.

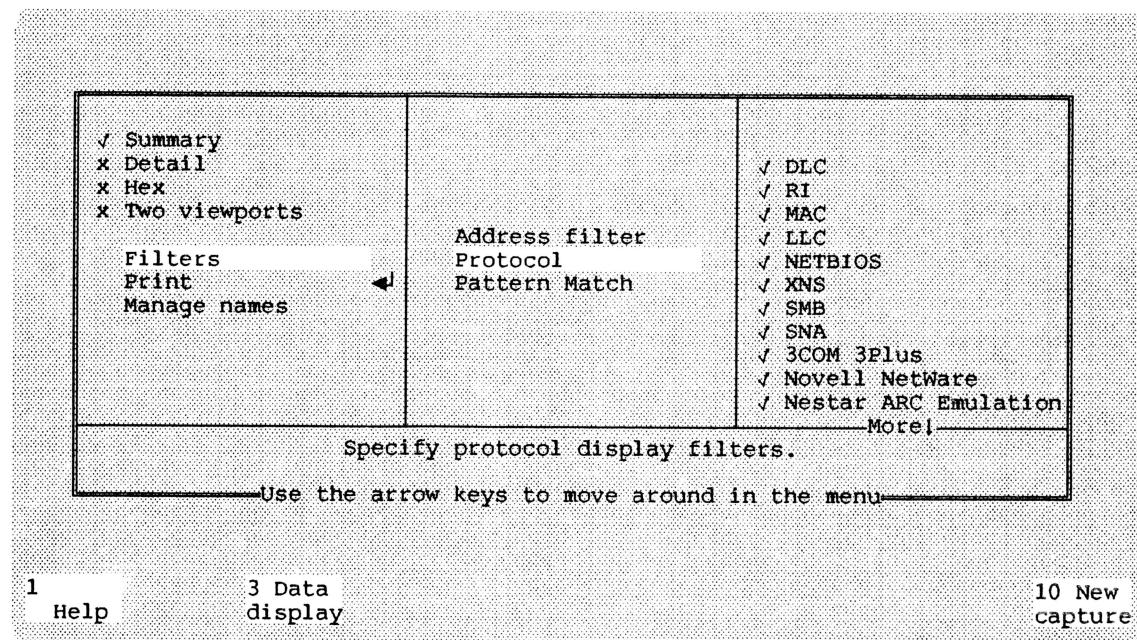


Figure 5-5: Display filters menu, showing list of protocol levels.

The display shows all the protocol levels that the Sniffer can interpret. That includes both those supplied by Network General and also any additional protocol interpreters you may have installed (see Appendix C). The category *Other SAP* selects a frame whose SAP is not covered by any of the other protocols on the list.

## Setting the Form of Display

There are three ways you can view a frame. You can have any or all views on the screen simultaneously. The three views are:

- **Hexadecimal view.** All bytes of the entire frame are shown, accompanied by a translation to ASCII or EBCDIC text.
- **Detail view.** The protocol is identified and standard fields within it are labeled and explained. Station addresses are replaced by their symbolic equivalents, according to the address table you supplied in file STARTUP.TRD. Since a low-level frame may contain higher-level frames within it, a single frame may require several levels of interpretation.
- **Summary view.** A short form of the hexadecimal and detail listings, condensed so that each level fits on a single line. You may elect to show all levels of interpretation, or only the highest.

### Hexadecimal view

The hexadecimal view shows each byte by its two-character description 00 to FF, with a blank between successive bytes. The bytes are arranged 16 to a row in a full-width table (8 to a row in a half-width table). At the left, the offset from the beginning of the frame is displayed in hexadecimal (Figure 5-6).

HEX	EBCDIC
0000 10 40 40 00 00 00 00 02	40 00 00 00 00 01 04 04 . ..... ....
0010 00 00 2D 00 01 01 00 0C	6B 80 00 31 00 13 07 B0 .....,.....
0020 B0 D0 B1 02 00 85 85 80	02 06 02 00 00 00 00 00 ..}..ee.....
0030 00 00 00 20 00 00 08 E2	C5 D5 C4 D3 E4 40 40 25 ..SENDLU ..
0040 00 09 02 D5 D6 D9 D4 C1	D3 40 40 09 03 00 00 00 ..NORMAL ..
0050 00 0D 00 00 00 0F 04 D5	C5 E3 E6 D6 D9 D2 4B E2 ..NETWORK.S ..
0060 C5 D5 C4 D3 E4 00 08 D9	C3 E5 D3 E4 40 40 40 ENDLU..RCVLU

Frame 35 of 225

1 Help 2 Set mark 5 Menus 6 Dispaly options 7 Prev frame 8 Next frame 10 New capture

Figure 5-6: Hexadecimal view of a frame.

To the right, the same characters are displayed again, this time with interpretable characters shown by their ASCII or EBCDIC equivalents, and all others by a dot. When you elect *Dynamic mode*

for the interpretation of hex characters, the Sniffer automatically selects EBCDIC to interpret MAC or SNA frames, and ASCII for all others. You can force ASCII or EBCDIC translation by selecting one of those alternatives instead (Figure 5-7).

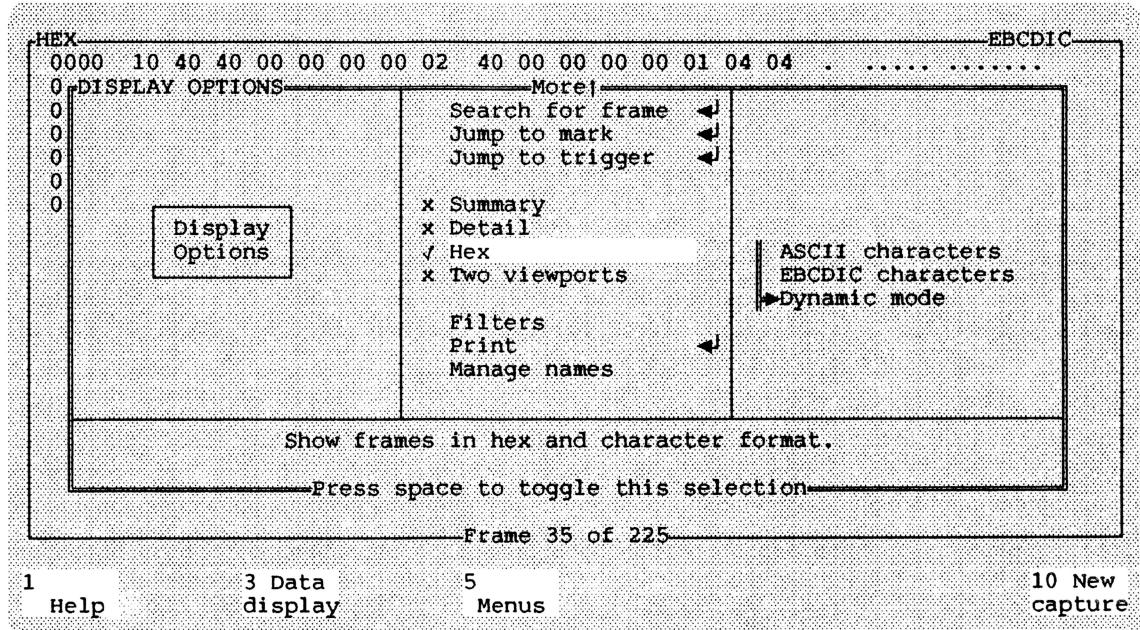


Figure 5-7. Menu to select the translation of hex characters.

## Detail view

The detail view first describes the outermost level of the frame. For example, the frame shown in Figures 5-8 and 5-9 is a data-link control (DLC) frame which contains within it a logical link control frame (LLC) which in turn contains within it an SNA frame. The first view of the frame shows the detail window automatically scrolled to include the beginning of the highest level protocol (Figure 5-8).

DETAIL	
SNA:	----- SNA Transmission Header -----
SNA:	
SNA:	Format identification (FID) = 2
SNA:	
SNA:	Transmission header flags = 2D
SNA:	0010 .... = Format identification is type 2
SNA:	.... 11.. = Only segment
SNA:	.... ..0. = Address field negotiation flag
SNA:	.... ...1 = Expedited flow
SNA:	Reserved = 00
SNA:	Destination address field = 01
SNA:	Origin address field = 01
SNA:	Sequence number = 12
SNA:	
SNA:	----- SNA Response Header (RH) -----
SNA:	
SNA:	Response flag byte 0 = 6B
SNA:	0... .... = Command
SNA:	..1. .... = RU category is 'data flow control'
SNA:	...0 .... = Reserved

Frame 35 of 225

1 Help    2 Set mark    5 Menus    6Display options    7 Prev frame    8 Next frame    10 New capture

Figure 5-8: Part of the **Detail** view of the frame that was shown in hexadecimal in Figure 5-6.

At the left edge of the detail view, the Sniffer shows the protocol level it's interpreting. In Figure 5-8, the interpretation is at the SNA level. When you've been working with *Highest level only* selected, the detail view initially presents the frame scrolled so so that the window starts with the highest level showing. If the detail interpretation is lengthy, other parts of it may not be visible. But you can use the up or down arrows, or the page keys, to scroll to other parts of the detail view. Figure 5-8 shows the detail view including the start of the SNA interpretation of that frame. By scrolling upward, you reveal the DLC and LLC interpretations that precede it (Figure 5-9).

```

DETAIL
DLC: ----- DLC Header -----
DLC:
DLC: Frame 35 arrived at 17:18:27.576; frame size is 006F (111 decimal) byt
DLC: AC: Frame priority 0, Reservation priority 0, Monitor count 0
DLC: FC: LLC frame, PCF attention code: None
DLC: FS: Addr recognized indicators: 00, Frame copied indicators: 00
DLC: Destination: Station 400000000002, Harrys PC
DLC: Source : Station 400000000001, Newman
DLC:
LLC: ----- LLC Header -----
LLC:
LLC: DSAP = 04, SSAP = 04, Command, I-frame, N(R) = 0, N(S) = 0
LLC:
SNA: ----- SNA Transmission Header -----
SNA:
SNA: Format identification (FID) = 2
SNA:
SNA: Transmission header flags = 2D
SNA: 0010 .... = Format identification is type 2
SNA: .... 11.. = Only segment
Frame 35 of 225

```

1 Help    2 Set mark                        5 Menus    6Display options    7 Prev frame    8 Next frame    10 New capture

Figure 5-9: Scrolling reveals other levels of detail in the same frame.

In the detail view of the DLC level, the frame's size, source, destination, and arrival time at the Sniffer, are all shown, together with various other control indicators contained within it.

The data transmitted by the DLC frame constitute an LLC (logical link control) frame, which is interpreted next.

The data transmitted with the LLC frame in turn contains a SNA frame, whose interpretation follows.

## Address Recognized and Frame Copied Bits

At the end of each frame as it circulates on the ring is a single byte which is used to check the frame's validity. Strictly speaking, it isn't part of the frame, and so it isn't included in the hex listing. However, the detail view reports the status of two bits within that trailer byte: *address recognized* and *frame copied*. They're visible in Figure 5-9.

The address-recognized bits indicate that at least one station upstream from the Sniffer has recognized itself in the frame's addressee field. The frame-copied bits indicate that the station was also able to make its own copy of the frame addressed to it. Normally, when the addressee both recognizes and records the frame, all four bits are set at the same time. Note that whether these bits have been set when the Sniffer sees the frame depends on where the Sniffer is located on the ring with respect to the frame's sender and the stations that might recognize or copy it. You may wish to take advantage of this by connecting the

Sniffer's cable so that you can see whether frames of interest are accepted by the station to which they're addressed.

## Displaying Both Detail and Hexadecimal Listing

When you select two or three views at once, the Sniffer divides the screen into two or three windows, one above the other. In such multi-window views, the summary window appears *above* any other windows, and the hexadecimal window appears *below* any other windows (Figure 5-10).

**DETAIL**

```
SNA: ----- SNA SC-RU (Session Control Response Unit) -----
SNA:
SNA: SC code = 31 (BIND: Bind Session)
SNA: Format/type flags = 00
SNA: FM profile flags = 13
SNA: TS profile flags = 07
SNA: Primary LU protocol flags = B0
SNA:           1... .... = Multiple RU chains allowed from primary LU
SNA:           .0... .... = Immediate request mode
Frame 35 of 225
```

**HEX**

			EBCDIC	
0000	10	40	40	00 00 00 00 00 02
0010	00	00	2D	00 01 01 00 0C
0020	B0	D0	B1	02 00 85 85 80
0030	00	00	20	00 00 08 E2
0040	00	09	02 D5 D6 D9 D4 C1	D3 40 40 09 03 00 00 00
0050	00	0D	00 00 00 0F 04 D5	C5 E3 E6 D6 D9 D2 4B E2
0060	C5	D5	C4 D3 E4 00 08 D9	C3 E5 D3 E4 40 40 40

Frame 35 of 225  
Use TAB to select windows.

1 Help    2 Set mark    4 Zoom in    5 Menus    6 Display options    7 Prev frame    8 Next frame    10 New capture

Figure 5-10: Detail and hexadecimal views of the same frame, shown in two windows.

## Summary View

The summary view shows several frames on either side of the one you've currently selected. It abbreviates and condenses so that each level of description is forced into a single line. The summary thus permits you to see at a glance the sequence and context of the frames, even though the individual descriptions are truncated. You can then examine individual frames in greater detail, or skip over them, as you wish.

SUMMARY	Delta t	DST	SRC	
25		Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
26	0.010	Harrys PC	+Newman	SNA XID Fmt 3 T2 NETWORK .SEND
27	0.013	Harrys PC	+Newman	SNA XID Fmt 3 T2 NETWORK .SEND
28	0.009	Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
29	0.016	Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
30	0.008	Harrys PC	+Newman	SNA XID Fmt 3 T2 NETWORK .SEND
35	0.083	Harrys PC	↔Newman	SNA REQ: BIND SENDLU
37	0.061	Newman	+Harrys PC	SNA RSP: BIND
39	0.044	Harrys PC	↔Newman	SNA REQ: LUSTAT
41	0.028	Newman	+Harrys PC	SNA RSP: FMD
43	0.021	Newman	+Harrys PC	SNA REQ: BIS
45	0.021	Harrys PC	↔Newman	SNA RSP: FMD
47	0.020	Harrys PC	↔Newman	SNA REQ: BIS
49	0.059	Newman	+Harrys PC	SNA REQ: UNBIND
58	0.123	Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
59	0.011	Harrys PC	↔Newman	SNA XID Fmt 3 T2 NETWORK .SEND
60	0.013	Harrys PC	↔Newman	SNA XID Fmt 3 T2 NETWORK .SEND
61	0.012	Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
62	0.014	Newman	+Harrys PC	SNA XID Fmt 3 T2 NETWORK .RCVP
63	0.011	Harrys PC	↔Newman	SNA XID Fmt 3 T2 NETWORK .SEND

1      2 Set      5      6Display      7 Prev frame      8 Next frame      10 New capture  
Help      mark      Menus      options

Figure 5-11: Summary view, showing frame 35 in the context of neighboring frames.

## Summary View in Two-Station Format

When you're studying the interchange between a pair of stations on the network, the pattern of dialog can be made clearer by arranging transmissions from one station on one side of the display, and those in the reverse direction on the other (Figure 5-12). You do that by electing *Two Station Format* for the summary view.

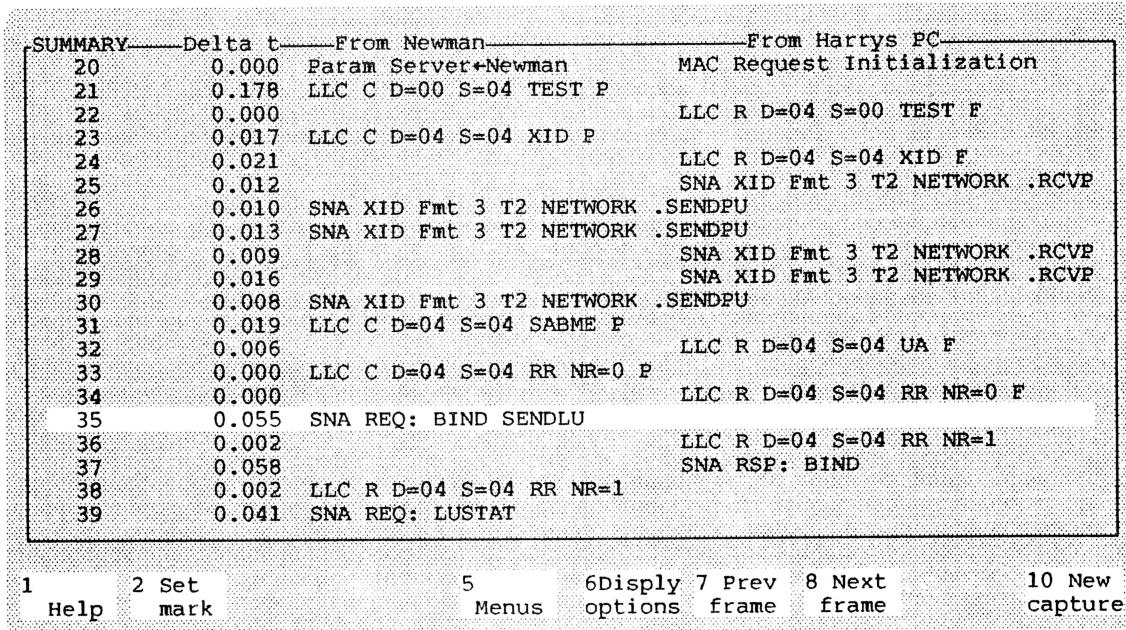


Figure 5-12: Two-station form of the summary view.

Transmissions involving just the two stations are shown half-width, those sent from one on one side, and those from the other on the other.

Frames to or from other stations, or frames sent with a specific destination (e.g., broadcast frames) remain in the usual full-width format. (In Figure 5-12, frame 21 to 39 are in the two-station format, whereas frame 20 is not.)

## How the Sniffer Knows Which Stations to Show in Two-Station Format

When you elect two-station format, the Sniffer scans the Capture Buffer for the first frame that is directed to a specific addressee. The station that sent that frame gets the left side of the display, and its addressee gets the right side.

If the first station-to-station message in the Capture Buffer involves some other pairing of stations, you should apply a display filter that limits the display to frames from the stations you want, and then switch to the two-station format.

## Multiple Levels with the Summary View

In the summary view (but not the others) by default the display is limited to the *highest* (that is, the most deeply embedded) level of interpretation. For example, a single DLC frame may contain within it an LLC level, which in turn contains an SNA level. With highest-level-only view, you see the SNA level, but not the DLC and LLC levels that contain it.

You can turn off highest-level-only view, and see a summary of each level within each frame. You do that as follows: select the *Display Summary* window, move the cursor so that *Highest Level Only* is highlighted, and press the space bar to remove the checkmark on that entry. (Pressing the space bar turns on the check mark when it's off, and turns it off when it's on.) The effect on summary display is shown in Figure 5-13.

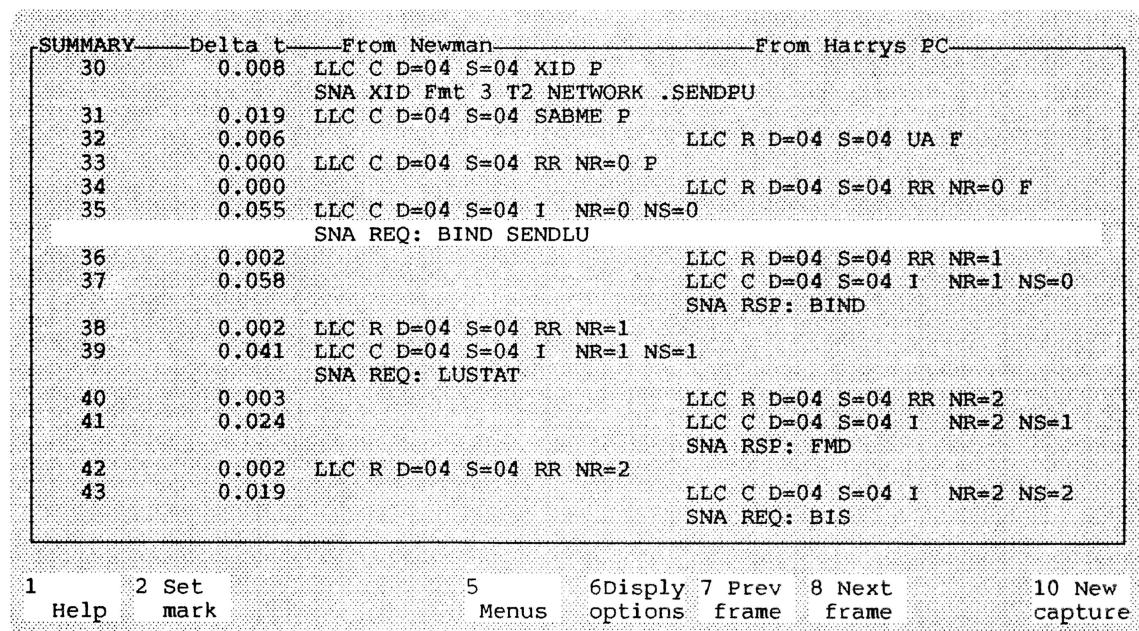


Figure 5-13: Summary display with the highest-level-only restriction removed (here shown in two-station format).

## Two Viewports Side-by-side

You may want to hold one frame on the screen while you examine another frame elsewhere in the Capture Buffer. To do that, you open a second viewport. In the *Display* menu, select *Two viewports*. The second viewport appears to the right. Each of the displays is truncated to half width in order to accomodate both sets.

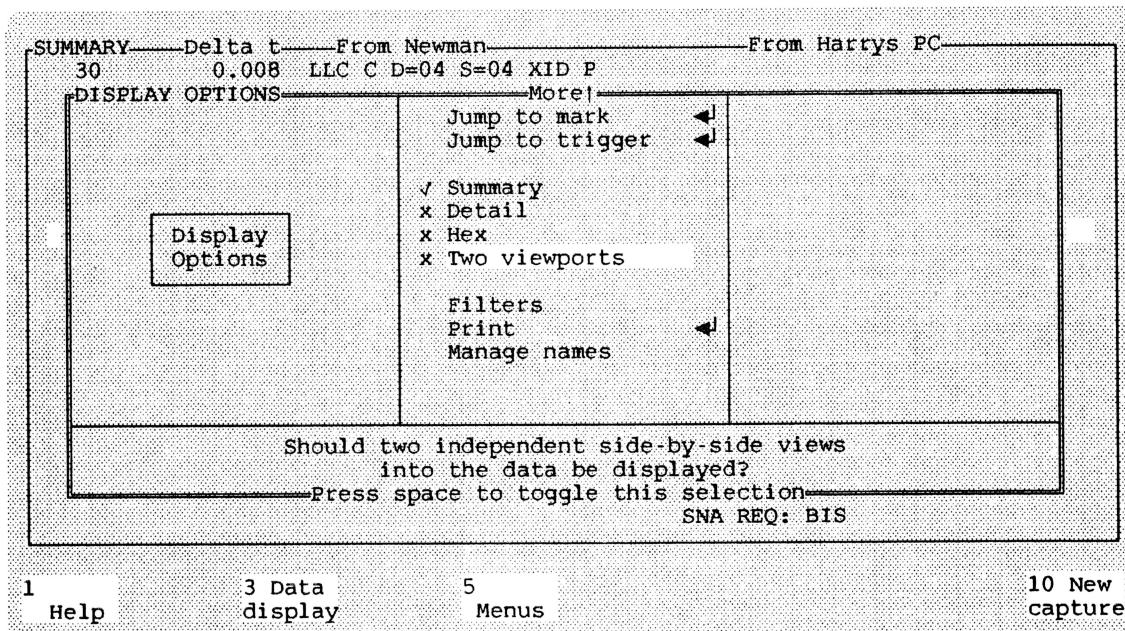


Figure 5-14: Menu to select two independent side-by-side viewports, superimposed summary display.

Figure 5-15 illustrates the use of two viewports to compare frames 35 and 49.

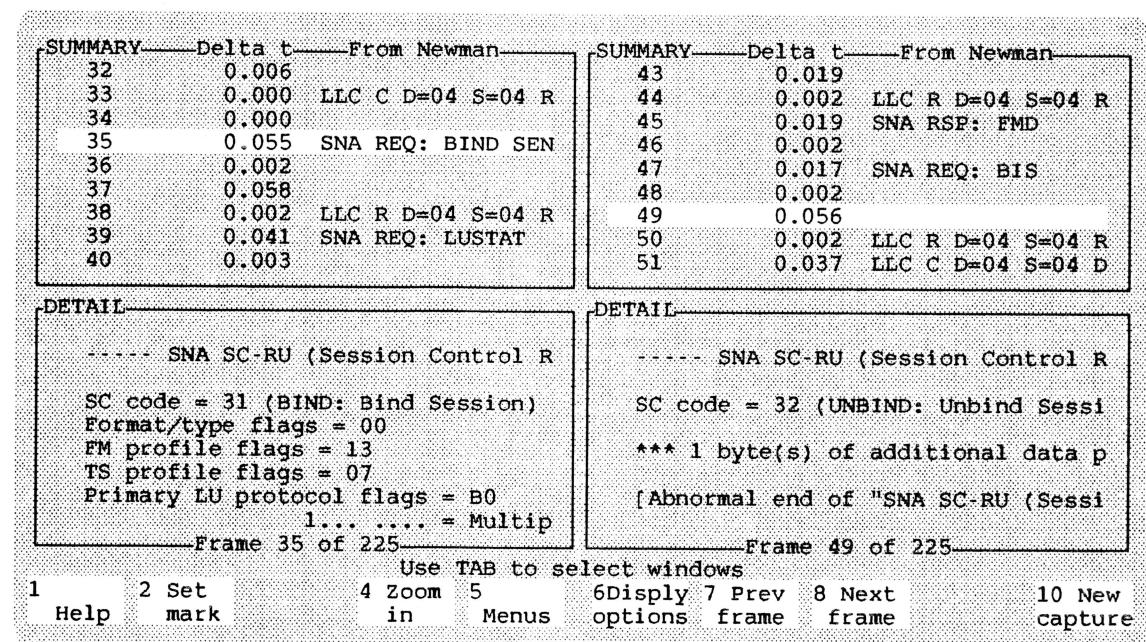


Figure 5-15: Display with two viewports, each containing a summary window and a detail window.

As you can imagine, using two viewpoints on a single screen, each containing multiple windows, doesn't leave very much room for any particular window. However, it's possible to take an enlarged look at one of the windows and then return to the multiwindow overview. See *Zooming for an Enlarged View*, later in this chapter.

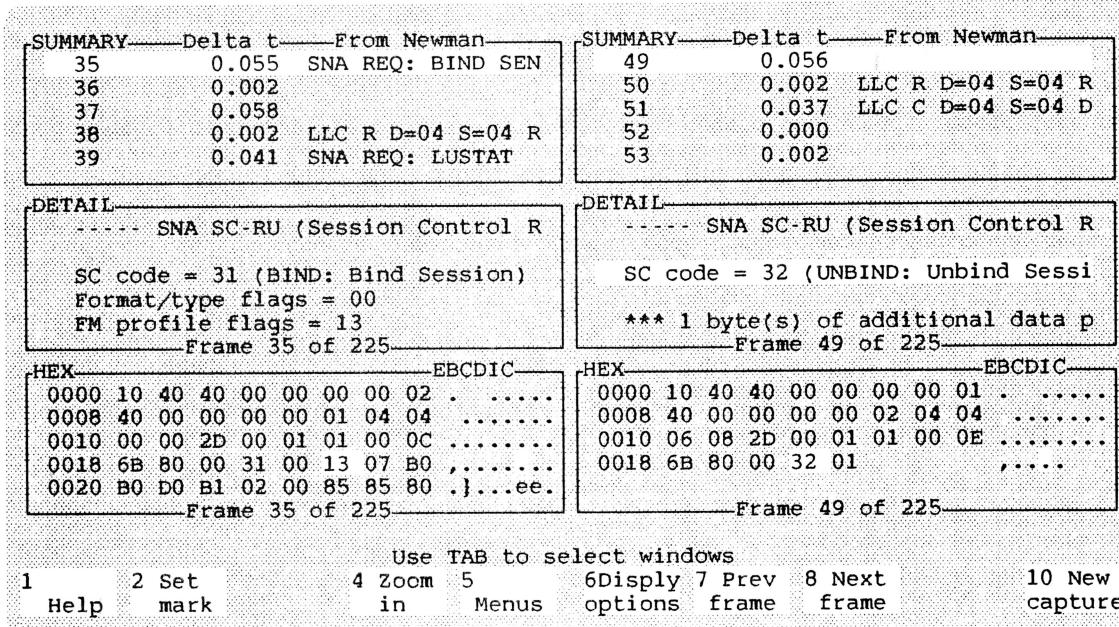


Figure 5-16: Two viewports, each with three windows.

## The Active Window

When the display contains more than one window, the window in which the highlight is located is considered the *active window*.

You move from one window to the next by pressing the *Tab* key. Each press takes you to the window below. When there are two viewports, going down from the lowest window on the left takes you to the top window on the right. *Shift Tab* moves you in the opposite direction.

Moving the highlight from one window to another makes the window at which you arrive the active window, and deactivates the window you left.

## **Highlighting Detail in the Hex Window**

When you have both detail and hex views of a frame, as you move the highlight to the various parts of the detail window, at the same time the Sniffer highlights the corresponding bytes in the hex window. That makes it easy to see the correspondence between the hexadecimal bytes that the frame actually contains and the interpretation placed upon them.

Since the hexadecimal window also displays the offset of each line, you can readily deduce the address of each field as you see it highlighted. (But note that if you use this information to supply an offset to the Sniffer's pattern matcher, offsets are there described as decimal values, while in the hex view the Sniffer shows them in hexadecimal.)

## **Scrolling Within a Window**

Especially when several windows are displayed at once, so that each is small, the information for a window may require more display space than the window contains. By using the movement keys (arrow, page, home, end, etc.) you can scroll the data within the active window, both vertically and horizontally. The up arrow moves you to frames that occur further up in the scroll, while the down arrow takes you to frames that occur further down. Similarly, the left arrow takes you to the left part of a frame, and the right arrow to the right part.

## **Scrolling to Next Frame**

Pressing **F7** takes you to the previous frame. Pressing **F8** takes you to the next frame. Only frames which pass the current display filter are shown, so pressing one of those keys takes you to the adjacent *visible* frame, perhaps skipping over one or more that the filter rejects.

In the detail window and the hex window, F7 and F8 replace the current display with the display for the neighboring frame. In the summary window, which can show several frames at once, F7 and F8 still move to the neighboring frame, but they do it by moving the zone that is highlighted, and scrolling if the highlight is already at the edge of the window.

While you're displaying two or three views, when you change to another frame, all the active viewport's windows scroll together. For example, when you're displaying both summary and detail, you might scroll the summary window so that it shows the next frame. That causes the detail window to show the next frame also, so that the windows remain in step.

Similarly, when you shift the highlight in the summary window to mark a different frame, the viewport's detail window and its hex window both scroll automatically, so that the frame they show remains the one that's highlighted in the summary window.

When you're showing multiple levels within the summary window, the report on a single frame may occupy several lines (one for each level). You can move the highlight from one level to the next, yet remain within the same frame. For example, you might move the highlight from the MAC level to the LLC level. When you do that, the detail window scrolls also, so that the level at the top of the detail window matches the level you've highlighted in the summary window.

### **Zooming for an Enlarged View of the Active Window**

Especially when you have several windows open, the space for the individual windows may be too small to see much of the information. You can temporarily expand the active window to fill the entire screen. Pressing **F4** zooms to full-screen display of the active window; pressing **F4** a second time restores the previous arrangement of windows.

### **Selecting the Focus of Display**

While you're displaying the Capture Buffer, the function key **F7** takes you to the previous frame, and **F8** takes you to the next frame. All views within a viewport (summary, detail and hex) move together.

When your active view is a summary view, scrolling with the up or down arrows may take you to another level of the same frame, or may take you to the next frame. When scrolling takes you to another level of the same frame, your detail view of that frame (if you have one open) moves too. When scrolling takes you to another frame, your hex or detail windows (if open) move to that frame too.

## Moving Directly to Another Frame

During display, if you press **F6** the superimposed menu of *Display Options* includes several ways of moving rapidly to a particular frame.

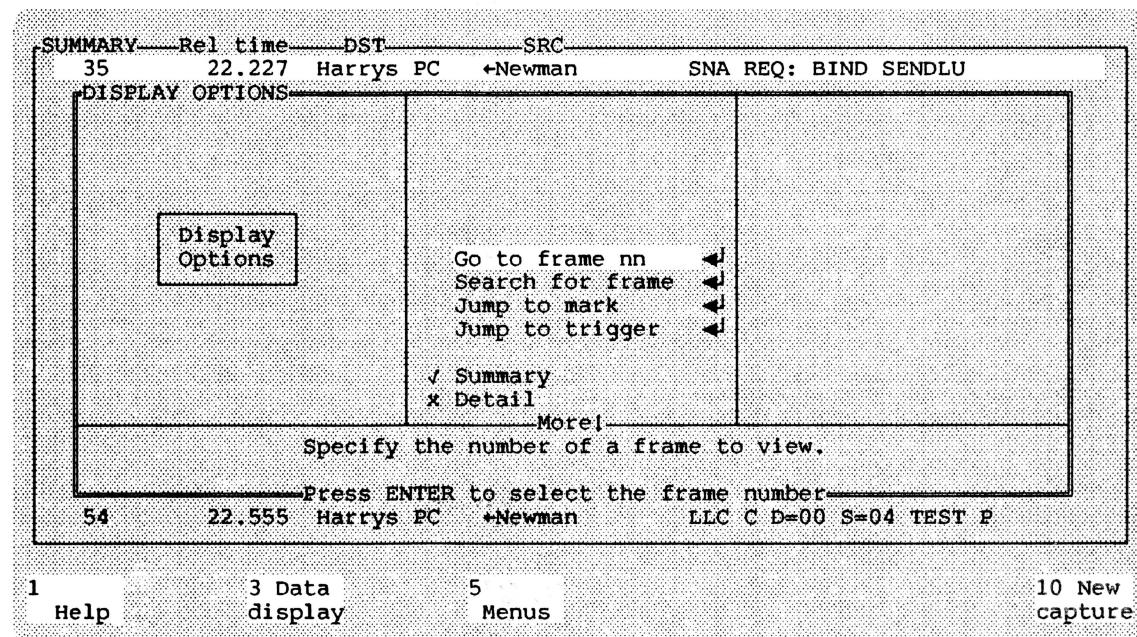


Figure 5-17: Superimposed menu showing options for moving around in the Capture Buffer.

*Jump to Trigger* takes you directly to the frame which was the trigger during capture, if there was one. (If present, it's marked with T in the summary view.) *Jump to Mark* takes you to the frame you marked with an M, or to the first frame if there is none marked. (You can set the mark on the current frame by pressing F2.)

## Jumping to a Frame Number

To move directly to a frame whose number you know, select *Go to frame nn*. When you press *Enter*, the Sniffer opens an additional window in which you can write the number of the frame you want. It won't let you ask for a frame is larger than the number of frames in the buffer (Figure 5-18).

The screenshot shows a terminal window displaying a list of network frames. The columns are labeled 'SUMMARY', 'Rel time', 'DST', 'SRC', and 'Protocol'. The frames are numbered from 35 to 54. Frame 35 is a SNA REQ: BIND SENDLU from Harrys PC to Newman. Frame 44 is a prompt 'ENTER FRAME NUMBER' with the value '134' entered. A message box is overlaid on the screen, containing the text 'Enter a frame number from 1 to 225' and 'Or press ESC to abort'. The bottom left corner of the terminal window has a small menu with options '1' and 'Help'.

SUMMARY	Rel time	DST	SRC	Protocol
35	22.227	Harrys PC	→Newman	SNA REQ: BIND SENDLU
36	22.230	Newman	→Harrys PC	LLC R D=04 S=04 RR NR=1
37	22.288	Newman	→Harrys PC	SNA RSP: BIND
38	22.291	Harrys PC	→Newman	LLC R D=04 S=04 RR NR=1
39	22.333	Harrys PC	→Newman	SNA REQ: LUSTAT
40	22.337	Newman	→Harrys PC	LLC R D=04 S=04 RR NR=2
41	22.361	Newman	→Harrys PC	SNA RSP: FMD
42	22.364	Harrys PC	→Newman	LLC R D=04 S=04 RR NR=2
43	22.383	Newman	→Harrys PC	SNA REQ: BIS
44				=04 RR NR=3
45				=04 RR NR=3
46				=04 RR NR=3
47				=04 RR NR=4
48				IND
49				=04 RR NR=4
50				
51	22.524	Harrys PC	→Newman	LLC C D=04 S=04 DISC P
52	22.525	Newman	→Harrys PC	LLC R D=04 S=04 UA F
53	22.527	Newman	→Harrys PC	LLC R D=04 S=04 DM
54	22.555	Harrys PC	→Newman	LLC C D=00 S=04 TEST P

Figure 5-18: Superimposed screen on which to write the number of the frame to which you want to go.

## Jumping to a Frame that's Filtered Out

When the frame you ask for (by number, or because it's the marked or trigger frame) is present in the Capture Buffer but excluded by your current display filters, the display moves to the first visible frame after it, or, failing that, to the first visible frame before it.

## Searching for a Pattern

You can move to a frame containing a particular pattern. You set the pattern in exactly the same way you'd state it for a capture filter or trigger pattern (described in Chapter 4).

Figure 5-19 shows a pattern to locate *ring purge*. Of course, a pattern makes sense only in the context of a protocol. To confine your search to frames in which that pattern in fact means *ring purge*, you have to set the display filter to pass only MAC frames.

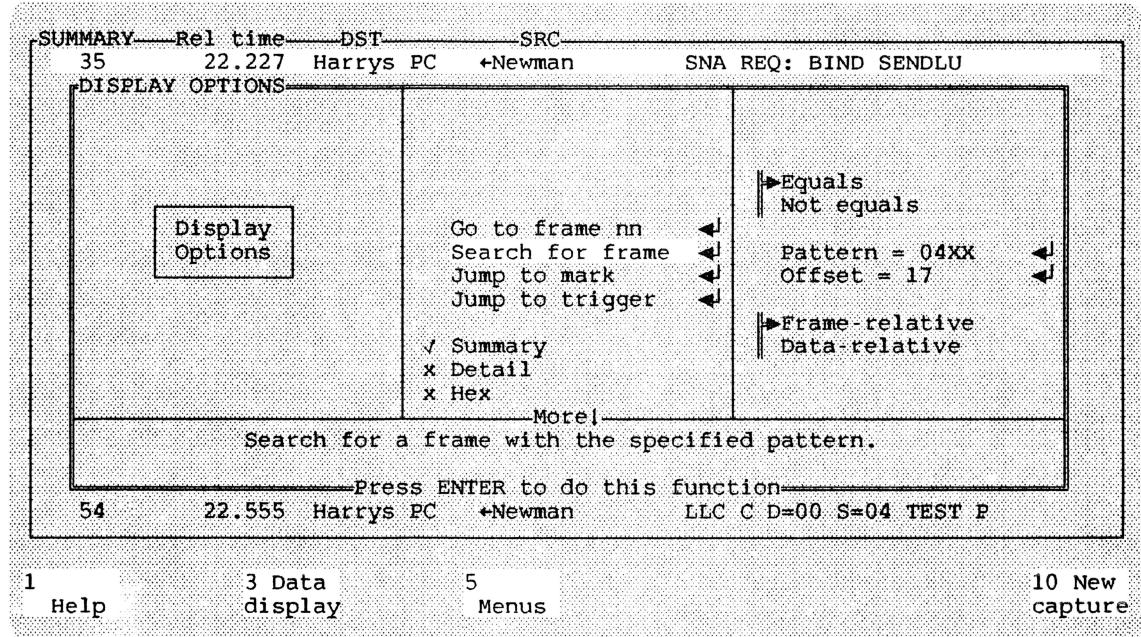


Figure 5-19: Specifying a pattern to jump to.

## How Time Is Displayed

The timing of network transmission is crucial to the examination of throughput. In its summary display, the Sniffer provides several alternative measures of time, selectable to match the situation. They're as follows:

- **Absolute time.** When the Sniffer completes reception of a frame, it attaches a timestamp. The timestamp records the time according to the Sniffer's internal clock at the moment the Sniffer recorded the end of the frame. All of the displays of time are computed from the absolute value recorded with each frame. Absolute time is displayed as hours, minutes, and seconds to the nearest millisecond. (That's also how time is shown in the detail display.)
- **Delta time.** Under this option, the time shown is the interval from the arrival of the preceding selected frame, in seconds to the nearest millisecond. Note that because it's the interval to the preceding *selected* frame, frames present in the Capture Buffer but not displayed don't affect delta time.
- **Relative time.** The time shown is the difference (in seconds, to three decimal places) between the frame's timestamp and the timestamp of the reference frame. The

reference frame is marked by a letter M appearing to the left of the frame number. When you first display the Buffer, the first frame is marked. You can mark a frame (and thereby remove the mark from any other frame) by pressing F2 (Set Mark) while you have the frame highlighted.

- **Network Utilization.** The number shown is an estimate of the percentage of the network's bandwidth (4 Mbits/sec) devoted to transmitting each frame now visible in the Capture Buffer, during a time interval that brackets the frame.

You can set the size of the bracketing window around each frame. The allowable sizes are 1, 10, 100 or 1000 milliseconds (Figure 5-20). For example, if you pick 100 milliseconds, the value reported is the number of bits in all the frames whose arrival times are within  $\pm 50$  milliseconds of the arrival time of the frame you're looking at, divided by the maximum number of bits that could be transmitted in that time, stated as a percentage. It's thus a moving average. When you pick a small window, you'll see larger momentary fluctuations; a larger interval smooths them out.

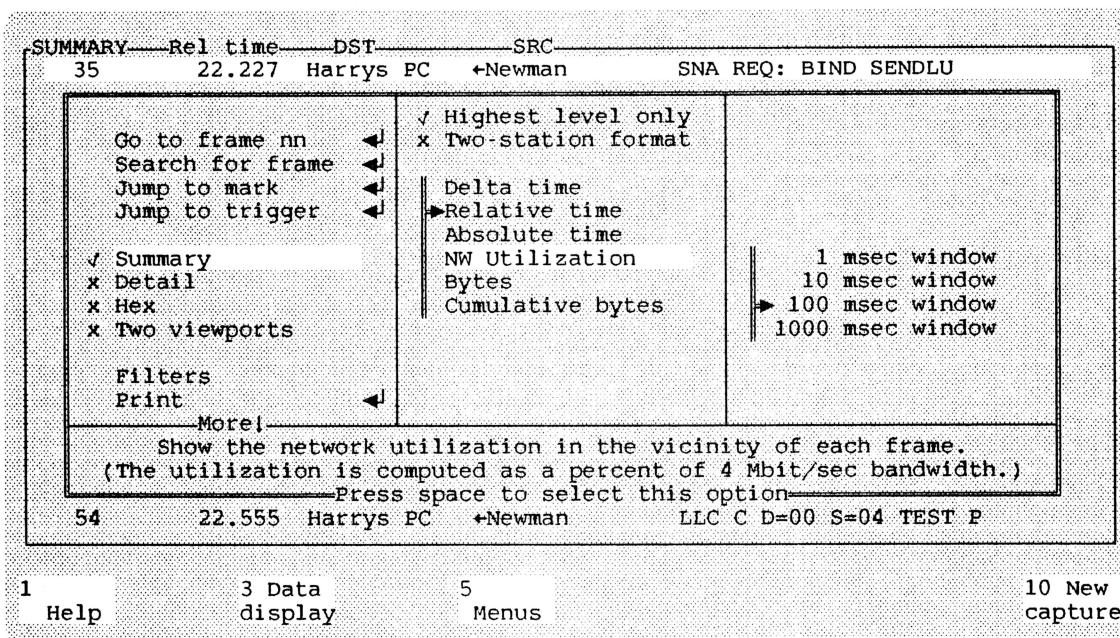


Figure 5-20: Menu to select the form of time display, superimposed on the summary window, showing additional options for average network utilization.

The estimate of network utilization is based on a sampling only of the frames that pass your capture and display filters. To make sense of utilization, it's important (a) to *exclude* unrelated frames (for example, from other stations), and (b) to *include* lower-level protocols that support a high-level transaction (for example, LLC or NETBIOS frames that support an SMB exchange).

## Managing Names Used in Displays and Filters

To make its displays more readable, the Sniffer substitutes symbolic station names for the hexadecimal station addresses. (In detail views, it shows both the hexadecimal address and a symbolic equivalent.) To translate between station addresses and the names that appear in the display, the Sniffer refers to a table of name definitions. When you start the Sniffer, it initializes that table by reading from the file of name definitions called STARTUP.TRD.

As it displays frames from the Capture Buffer, the Sniffer checks every station address against its name table. Whenever it encounters an address that's not in the table, it adds an entry to the table. The entry contains the hexadecimal address, but with a blank symbolic name. The Sniffer adds both source addresses and destination addresses. Since some messages are addressed to a group or a function rather than to a station (for example to *Error Monitor*, or *Broadcast*), the table will include the hexadecimal addresses of those functions as well as the addresses of stations that transmit.

There are several ways you can manage the Sniffer's use of names. From the *Files* option in the main menu, select *Manage Names*, and from there one of several alternatives (Figure 5-21).

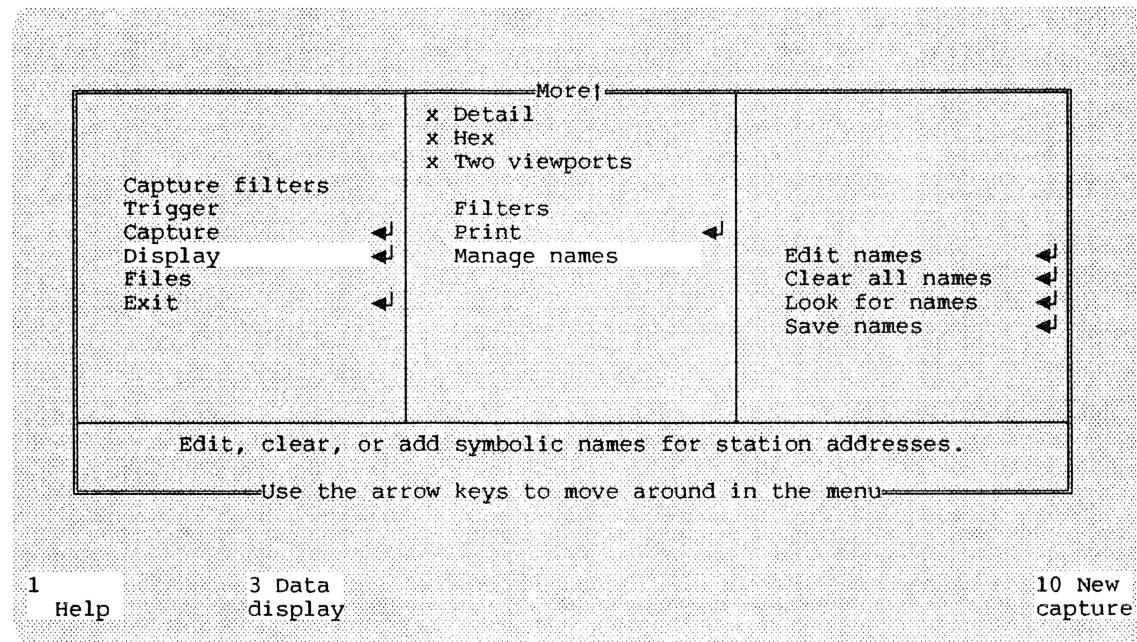


Figure 5-21: Menu options for managing names used in Sniffer displays.

In general, you manage names by editing the working name table. Changes to the name table don't affect the file STARTUP.TRD (and hence don't last longer than your current work with the

Sniffer) until you select *Save Names*, which replaces what's now in the file STARTUP.TRD with the current working names table.

### **The Sniffer Assigns a Name for Itself**

When it first inserts itself into the ring, the Sniffer inserts a name for itself in the names table. It calls itself "This Sniffer." (It does that even if its station address was initially assigned some other name.)

### **Deducing What the Stations Call Themselves**

Many network systems permit machines to assign symbolic names to themselves. When you select the option *Look for Names*, the Sniffer searches through frames in the Capture Buffer in search of symbolic names embedded in the transmissions there. However, it looks for names only for those stations which don't already have names in its name table. If you want to know what the stations are (for the moment) calling themselves, it's best to run *Look for Names* before you start assigning names yourself. You can select *Clear Names* to remove all names from the table.

Names that the Sniffer finds in this fashion may be quite transitory. For example, you may find a name that a particular user assigned solely for a particular work session. The user may move to another machine and reassign the same name somewhere else. Because such names are so readily changed, you may not want to save a name table constructed this way, since it may be wrong next time you use it.

## Editing the Names Table

When you select *Edit names*, the Sniffer displays the current names table (Figure 5-22). One line in the table is highlighted. You can scroll through the table to move the highlight, or bring into view names that didn't fit in the window.

The screenshot shows a terminal window titled "EDIT NAMES". The window contains a table of network names and their corresponding MAC addresses. The table is as follows:

	NAME	MAC ADDRESS
x	New station	C00000000001
x	Active Mon.	C00000000100
x	All bridges	C00000000100
Filte	Bridge #1	480002008592
Print	Bridge #2	480002008593
Manag	Bridge #3	480002008596
	Bridge #4	480002008598
	Broadcast	C000FFFFFF
	Error Monitr	C00000000008
	Groucho	42608C187065
	Harrys PC	400000000002
	IBM PC WS	10005A0035FC
	LAN Manager	C00000000010

Below the table, a message reads: "Use ! and | then press ENTER, or ESC to abort." At the bottom of the window, it says "Press ENTER to do this function".

In the bottom left corner of the terminal window, there is a small menu with the number "1" and the word "Help".

Figure 5-22: Display of the names table.

To edit a name, you scroll until the name you want to change is highlighted, and press *Enter*. The Sniffer opens a window in which you can write a new symbolic name for that address (Figure 5-23).

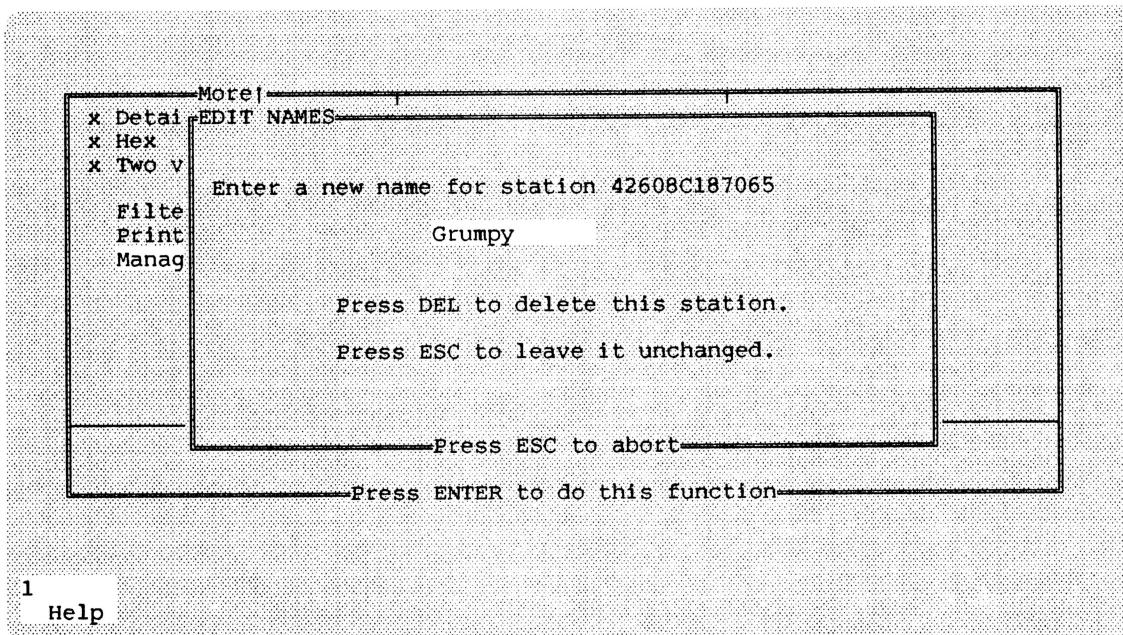


Figure 5-23: Window to provide a new symbolic name for a station.

When you edit the line headed *New Station*, the Sniffer asks both for a station address and a symbolic name. (You can also reach that window if you select *New Station* when setting a filter on station address.)

## Printing a Report on Frames in the Capture Buffer

You can obtain a printed record of the Sniffer's displays of the Capture Buffer. The report can be sent directly to a printer, or to a file (Figure 5-24).

A report sent directly to a printer may be routed either to LPT1 or to COM1. LPT1 is for a parallel connection, by way of a DB25 connector. COM1 is for a serial port, via a DB9 connector. If you use a serial port, you will also have to specify the rate (in baud) of communication, and the mode; see the discussion of these matters in the printer manual or the DOS manual.

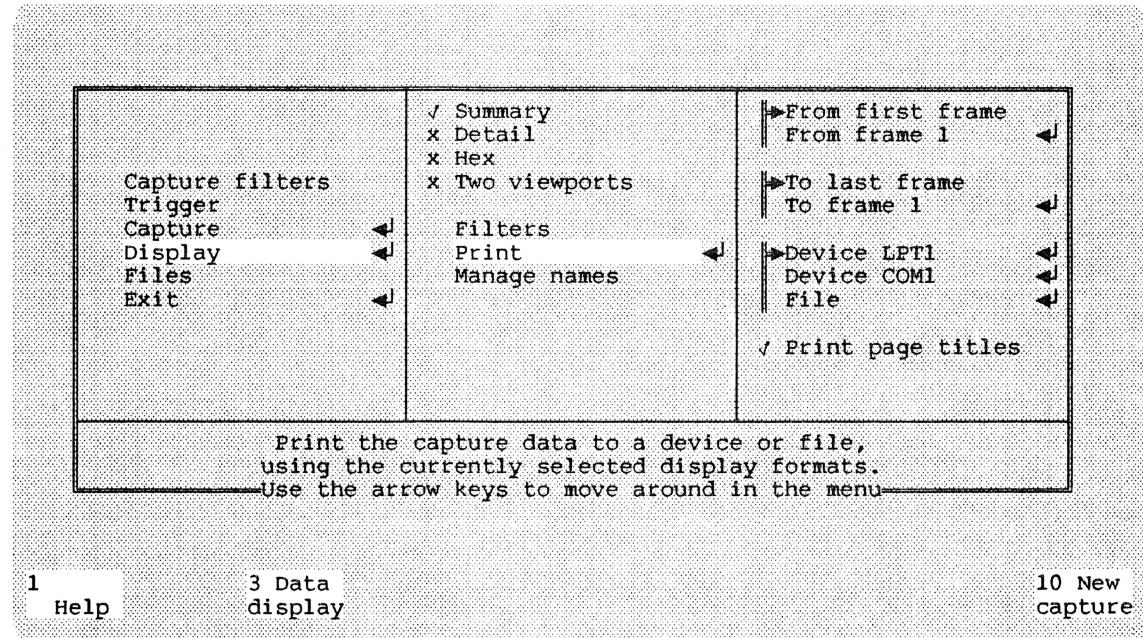


Figure 5-24: Menu to select printing of a report on frames in the Capture Buffer.

The printed report contains the same information you can see displayed on the screen, without the restriction of the small window, or the requirement to scroll. Note the following points of similarity and difference between the screen display and the printed version:

- The printed report starts with a record of the date and time, and the name of the file from which the Capture Buffer was loaded (when applicable).
- Like the screen display, the printed report shows each of the views you requested for a frame, then the same views for the next frame, and so on. However, the printer does not break views into the windows required for viewing on the screen.
- A frame may contain several levels of protocol. In the detail view, the printed version shows only those levels of protocol actually checked in your display filter, and levels other than the highest only when you turn off *Highest level only*. (By contrast, for any frame that the filter accepts, the screen permits you to scroll inside the detail window to look at other levels of protocol that may be there.)

To print all levels of the detail interpretation, you must turn off *Highest level only* **and** turn on all the protocol levels in the display filter.

- The printer menu permits you to print all the frames in the Capture buffer, or to state a first frame and a last frame to print only a portion of it.

## Saving the Capture Buffer to a File

At the start of this chapter there's a discussion of loading the Capture Buffer with data previously saved in a file. This paragraph tells you how the file was saved.

While you have frames in the Capture Buffer (either because you just captured them, or because you loaded them from a file), you can select Files and then Save. You can either save everything in the Capture Buffer, or you can save only those frames that pass your current display filter. You indicate your choice by selecting Data or Filtered Data, as appropriate (Figure 5-25).

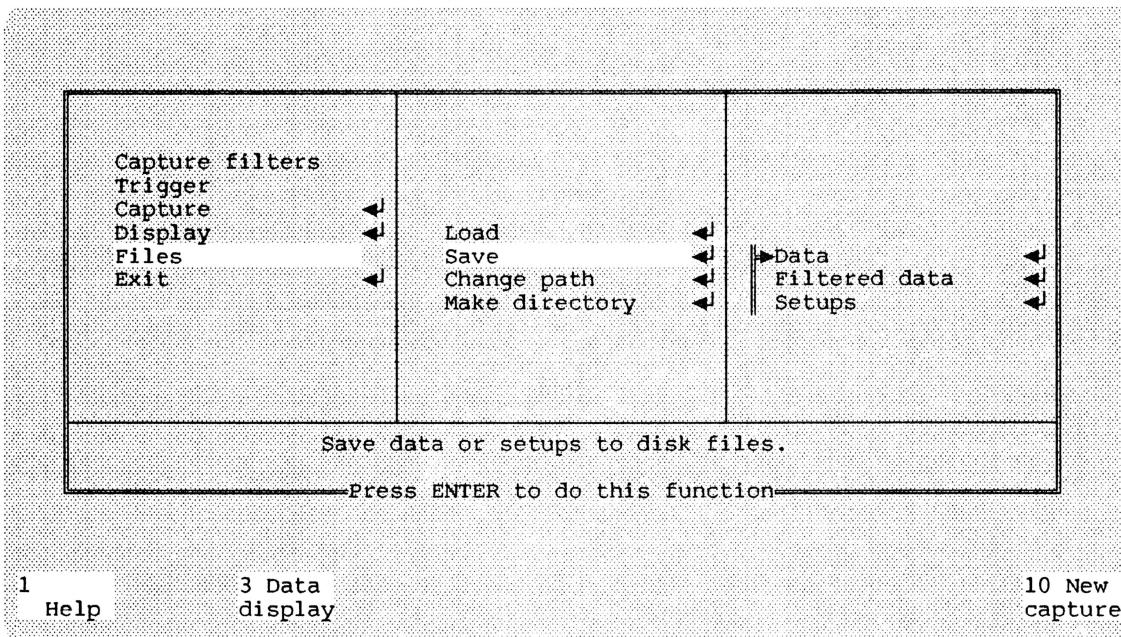


Figure 5-25: Menu for saving data files or setup files.

You can select which of the three save options you want by moving the highlight to the desired alternative, and then pressing the space bar to select it (and deselect the alternatives). If you press Enter while you have one of the three alternatives highlighted, you both select that alternative and start executing it.

The Sniffer prompts you for the name of the file to which you want your data written. If the file you name doesn't yet exist, the Sniffer creates it and writes it. If the file already exists, the Sniffer discards its former contents.

## Saving Your Current Setup

When you select *Files*, then *Save* and then *Setup*, the Sniffer asks you for the name of a file. In the file you name, the Sniffer writes a record of the your setup.

The setup file records what you had chosen for every one of the options settable in the menus by a checkmark or a radio control. (That includes capture options, filter options, trigger options, display options, printer options, and so on.)

The saved setup also records each of four separate patterns together with each pattern's offset, for the following uses:

- Capture filter
- Trigger
- Display filter
- Display search

The saved setup also records two pair of station names (the sending station and the addressee), one pair for each of the following uses:

- Capture filter
- Display filter

The saved setup does *not* record the path you may have set for locating the directory of files to be loaded or saved.

Saving the setup has no effect on saving the Capture Buffer; that's done independently by the option *Save, Data*.

Saving the setup has no effect on saving the name table; that's done independently by the option *Manage names, Save*).

## Using a Saved Setup File

When you first start the Sniffer, it always starts out with its own standard defaults. You activate a saved setup by selecting *Files*, then *Load*, then *Setup*. When the Sniffer shows you a list of saved setup files, select the one you want and press *Enter*. The Sniffer sets all options the way they were when you saved the setup file.



# Appendix A. Format of Saved Data Files

For some purposes of data analysis, it may be useful to write a program which reads the saved trace data files. This appendix describes the format of those files. Note, however, that in many cases it will be easier to have the program read the formatted file produced by “printing” one of the display formats to a file, particularly if you choose the option to omit page titles.

The data files saved by the Sniffer consist of sequences of variable length binary records. Since all 256 possible byte values are possible within the data, the file may not be edited by an ordinary text editor.

The first sequence of bytes in a data file is a text message (“**TRSNIFF data**”) ending with an endfile character (0x1A, Ctrl-z). This has been done so that even if you accidentally type the file to the screen, or otherwise treat it as a text file, the display reaches a terminator before reaching unprintable characters that may be in the data.

All multibyte arithmetic fields (computed by the Sniffer during capture) are stored with the least significant byte first. Frame data are stored in the byte order transmitted.

## Header

Following the text message string, the file contains an arbitrary number of variable-length records, each of which begins with the following header:

```
struct f_rec_struct           Standard record header.  
{  
    int   type                Type of this record.  
    int   length;             Length of remainder of this record.  
    int   rsvd;               Reserved word, currently 0.  
};
```

The first field of the record header indicates what type of record data follows, and currently can have the following values:

#define REC_VERS 1	Version record (f_vers).
#define REC_FRAME2 4	Frame data (f_frame2).
#define REC_EOF 3	Endfile record (no data follows).

Other types are reserved for future use.

## Structures within the Data File

The data file consists of a single *version* record, a series of *frame data* records, and a single *endfile* record.

The format of the data part of these records follows.

### Version Record

```
struct f_vers_struct {

    int maj_vers;                      Major version of Sniffer.

    int min_vers;                      Minor version of Sniffer.

    struct date_struct date;          Date & time (4 bytes, DOS format)

    int type;                          What type of records follow.

    int format;                        What format version this is.

    int rsvd[3];                      Reserved words.

};
```

## Frame data records.

```
struct f_frame2_struct {  
  
    int   time_low;           Low time in 0.838096 usec units.  
    int   time_mid;          Mid time in 54.9255 msec units.  
    int   time_high;         High time in hours.  
    int   size;              Number of bytes actually written in this file  
                           (may be less than the original length of the  
                           frame).  
    char  fs;                Saved FS byte of frame.  
    char  flags;             Buffer flags - for internal use.  
    int   true_size;         If non-zero, the size of the original frame (since  
                           not all the data was recorded).  
    int   rsvd3;              Reserved; currently 0.  
                           The frame data follows.  
};
```

## Endfile Record

The endfile record has no data; it consists only of the record header. There is no explicit encoding of the length of the file except as part of the file's directory entry.



## Appendix B. File Name Conventions

The Sniffer's executable code is named TRSNIFF.EXE.

At startup, the Sniffer initializes its names table by reading the file STARTUP.TRD from the current directory. You can update this file by saving the current names table, but you do not control the name of the file.

During execution, when you press **F1** to request help, the Sniffer refers to the file TRSNIFF.HLP. The help file should either reside in the current directory, or else in a directory locatable by including in the Sniffer's Autoexec file or the batch file that starts the Sniffer the DOS command `SET TRHELP = path`.

**Files of Captured Frames** You can save the capture buffer to a file that you name. The Sniffer accepts only a name without an extension, and supplies the extension .TRC for each such file. To load the capture buffer, you must select from a list the Sniffer displays; it considers only files with the extension .TRC.

You can save a record of the settings of all menu options to a file that you name. The Sniffer accepts only a name without an extension, and supplies the extension .TRS for each such file. To load the setup file, you must select from a list the Sniffer displays; it considers only files with the extension .TRS.

You may direct a printed record of the capture buffer to a file that you name. The Sniffer prompts for a name without an extension, and provides the extension .PRN.

Files with the extension .TRC, .TRS or .PRN are located in the current directory, or in the directory pointed to by the *Set path* option while the Sniffer is running.

**Reserved extensions** The following extensions are reserved for future development of the Sniffer:

- .TRI Screen images.
- .TAP Tutorial screens.
- .TRM Measurement files.
- .TRX Translate tables.



# **Appendix C. Extending Sniffer Protocol Interpreters**

## **Overview**

This appendix describes the rules and conventions for writing programs that extend the protocol interpretation of the Sniffer. The reader is expected to be familiar with:

- The general operation of the Sniffer
- The frame formats of the token ring Data Link Control and the IEEE 802.2 Logical Link Control protocol.
- The C programming language.

Network General is constantly expanding its suite of optional Protocol Interpreters. Before writing your own, check with us to see what is currently available.

A Protocol Interpreter ("PI") is a routine (or set of routines) which is given a pointer to data somewhere within a frame. The interpreter has four tasks:

- Generate one or more short text lines for its protocol level to be displayed in the summary window.
- Generate lines for the detail window.
- Call the Protocol Interpreters for embedded protocols, if any.
- (Optional) Supply new symbolic names discovered within the protocol data.

Protocols are initially demultiplexed by LLC DSAPs; any subsequent demultiplexing is done by Protocol Interpreters which are aware of specialized conventions for embedded protocols. Protocol Interpreters for embedded protocols may be shared; that is, a PI may be called from several other PI's.

## Calling conventions for Protocol Interpreters

A Protocol Interpreter function named `new_pi` will be called from the Sniffer as follows:

```
void new_pi (frame_ptr, frame_length)
char *frame_ptr;                                Pointer to frame data
int frame_length;                               The length of the frame data starting at
                                                frame_ptr
```

The pointer to the frame data starts within the physical frame at the beginning of the data for the protocol, and thus skips previous protocol fields including source and destination addresses, routing indicators, LLC control, and any previously-embedded protocol headers.

Future versions of the Sniffer will permit you to truncate frames in storage (for example, recording no more than a certain number of bytes for each frame, and discarding the rest). When truncation is elected, `frame_length` will be the length of the stored (truncated) record, while the global integer `true_length` will indicate the frame's length before truncation.

For first-level ("SAP") Protocol Interpreters called from the SAP demultiplexer, `frame_ptr` is the address of the first byte of the information field immediately following the source and destination SAPs and the control field. The SAP Protocol interpreter will be called for any frame with an information field, such as I, UI, XID, TEST, or FRMR. It will not be called for frames which do not contain information, such as RR.

If the Protocol Interpreter needs access to DLC or LLC header fields, or the frame number, the following global static variables may be used:

```
char *dlc_header;                                Pointer to DLC header of the frame, starting
                                                with the AC field
char *llc_header;                                Pointer to the LLC header of the frame, starting
                                                with the DSAP field
int llc_type;                                    The type of the LLC frame; see the LLC_XXX
                                                macros in pi.h
int pi_frame;                                    The current frame number
```

Protocol interpreters for embedded protocols should be invoked using the same calling convention.

## Registering Protocol Interpreters

Protocol interpreters must be registered before they are used. All registration occurs in the function `initpi.c` which is called when the Sniffer is initialized.

Registering a Protocol Interpreter generates a pointer to a structure which holds data relevant to the interpreter. That pointer must be supplied in subsequent calls which generate screen output. The pointer should be saved in a static variable known to the interpreter.

<code>struct pi_data *register_pi (menu_title, ndsaps, dsaps, pi, prefix)</code>	
<code>char *menu_title;</code>	A pointer to a string which will appear in the menu as a selectable item, which the user can use to control whether summary lines for that protocol should be displayed. It is also used in constructing the messages which appear at bottom of the menu window. The string should not exceed 18 characters.
<code>int ndsaps;</code>	The number of DSAPs which will be processed by this interpreter. If this is an embedded protocol, specify 0.
<code>int *dsaps;</code>	An array of “ndsaps” integers representing the DSAPs which will be processed by this interpreter. If this is an embedded protocol, this parameter is ignored and should be specified as a null address.
<code>void *pi();</code>	The address of the Protocol Interpreter function.
<code>char *prefix;</code>	A pointer to a string to be used as a prefix for lines in the detail window. For visual consistency with other Protocol Interpreters the string should be in the form “xxx: ”.

## The Protocol Interpreter Data Structure

This structure is allocated, and a pointer to it returned, by the `register_pi()` function. Only the fields described below are relevant to Protocol Interpreters, and they are booleans which indicate what functions are required. The integer booleans, in standard C fashion, are 0 if false and non-zero if true. The declaration for this structure is part of the file `pi.h`.

```
struct pi_data {
```

```
int do_sum;           Boolean: generate summary lines?  
int do_int;          Boolean: generate interpretation lines?  
int do_count;        Boolean: only count summary lines?  
int do_names;        Boolean: add symbolic station names?  
};  
If do_sum is true, generate summary lines. If do_count is also  
true, then only a count of summary lines is really needed, so for  
added efficiency you may optionally allocate the line buffers but  
omit actually generating the text.
```

If do\_int is true, generate detail interpretation lines.

If do\_names is true, the operator has selected the "Search for  
names" menu option. Examine the frame data for embedded  
station names defined by the protocol. If any are found, call the  
add\_station\_name() function, described later, to enter the names  
into the name table.

## **Generating output from Protocol Interpreters**

To generate a line for the summary window from within a Protocol Interpreter, first get the address of a line buffer by calling `get_sum_line()`:

```
char *get_sum_line (pid)
```

Returns a pointer to the line buffer

```
struct pi_data *pid;
```

The value returned when the interpreter was registered

Then move a character string, ending with a null, into the buffer provided. The length of the string including the null cannot exceed `MAX_SUM_LINE`. For visual consistency of the displayed output, the string should begin with a 3-character identification of the protocol layer.

Generating a line for the detail window is similar, except the function to get the address of a line buffer also provides an optional offset and length of the field within the frame that produced the information. This will be used to highlight the hex field of the frame when the corresponding detail line is selected.

```
char *get_int_line (pid, offset, length)
```

Returns a pointer to the line buffer

```
struct pi_data *pid;
```

The value returned when the interpreter was registered.

```
int offset;
```

The offset from the DLC header of the field which generated the interpretation line.

```
int length;
```

The length of the field, or 0 if no highlighting is desired.

The string should be built in the supplied buffer. Its length including the null must not exceed `MAX_INT_LINE`.

The detail window will normally be automatically scrolled so that the top line is the first line for the protocol which is selected in the summary window. If you wish some other line of your protocol's detail lines to scroll to the top, you may so indicate by using a negative offset when calling `get_int_line` for that line.

Summary or detail display lines should be generated only if the appropriate boolean in the `pi_data` structure is true. Embedded Protocol Interpreters should be invoked whether display lines are generated or not.

## **Adding Symbolic Names to the Name Table**

To add a symbolic name to the name table if do\_names is true, call the following function:

```
add_station_name (addr, name, replace)

int    addr [3];           The 6-byte station address.

char   *name;             A pointer to the new symbolic name, 1 to 12
                           characters followed by a NUL.

int    replace;            A boolean indicating whether an existing name
                           should be replaced.
```

Since names discovered in the protocol data may be transitory, it is probably best to make the third argument 0 (false) so that if a name was already supplied for that station address in the startup.trd file it will not be replaced.

## **Using existing Protocol Interpreters**

The existing Protocol Interpreters for NETBIOS (name management), SMBs (PC LAN Program protocol) and SNA (LU 6.2, APPC) may be called from newly-added Protocol Interpreters that have those protocols embedded. The calling sequence for each interpreter is as described above.

The names and calling structure of the existing Protocol Interpreters are as follows:

```
interp_dlc ()

interp_ri ()

interp_mac ()

interp_llc ()

interp_netbios ()      [for SAP 240]

interp_smb ()

interp_sna ()          [for SAPs 4, 5, 8, and 12]

interp_othersaps ()
```

Note that if the `interp_netbios()` Protocol Interpreter is invoked, it will in turn invoke the `interp_smb()` Protocol Interpreter for embedded SMBs.

## Dependencies on other frames

In cases where the interpretation of a frame must depend on information in prior frames, there are two mechanisms available:

- The PI can cache, in private static variables, any information that will be useful for subsequent frame interpretation. Beware, however, that the PI may be called to interpret frames in any order. The cache, therefore, will only sometimes be valid.

In order to recognize that cached data is invalid when new frames have been loaded or captured, the PI can examine the global integer variable `data_version` which will be incremented each time new data is present.

- The PI can get access to the data in other frames by calling the following routine:

```
int      pi_get_frame (frame_num, p_dlc, p_llc, p_data)
int    frame_num;           The desired frame number. The first frame is
                           frame number 0.
char   **p_dlc;            The address of a pointer which will be set to the
                           start of the DLC header.
char   **l_llc;            The address of a pointer which will be set to the
                           start of the LLC header.
char   **p_data;           The address of a pointer which will be set to the
                           start of the LLC data.
```

The return value will be the size of the frame if it is available. The length will be zero if the frame does not exist, or is not an LLC frame, or the DSAP is not one of the SAPs handled by the current PI.

Note that this technique requires the PI to know the offset of the its relevant data from the start of the LLC data; in the case of embedded PI's this requires knowledge of earlier PI's data structures.

## **Appendix D. A Brief Summary of the Token-Ring Network Architecture**

The Token Ring is a Local Area Network suitable for high speed interconnection of computers and computer-controlled devices over moderate distances. The architecture of the Token Ring is defined de facto by implementations from IBM, Texas Instruments, and others, and de jure as ANSI/IEEE standard 802.5 and ISO/DIS standard 8802/5.

Most system implementations of the Token-Ring network also use at least a subset of a similarly standardized protocol for Logical Link Control, referred to as LLC and defined as ANSI/IEEE standard 802.2 and ISO/DIS standard 8802/2.

This appendix is a highly condensed summary of some features of the Token-Ring network and LLC protocols which are related to the use and understanding of the Sniffer. Terms and acronyms which appear here are used as part of the screen displays. See also Appendix E, which contains a list of acronyms, for related information.

### **Physical Interconnection and Speed**

Stations connected to the Token-Ring network are wired together physically in star-like fashion; each station uses one cable to attach itself to the nearest passive concentrator, or MAU (multiple access unit). The MAUs can themselves be linked together and may be separated by moderate distances. The number of stations and distance limitations depend on several variables including cable type, but typically one or two hundred stations can be interconnected into a single network segment using cables between stations and MAUs up to 300 meters long. The distance limitations can be overcome by using special line drivers or fiber optic cables. Networks of many hundreds or thousands of stations over large distances can be created using computers as bridges between network segments.

The connectors used to attach to MAUs are designed by IBM and are hermaphroditic so that any two may be joined; cable "extension cords" have the same connector on both ends. The cable which connects to a particular computer may use the hermaphroditic connector if there is enough panel space for the mating connector (about one inch by one inch) or may use a non-standard connector. The convention for personal computers is to use a DB-9 female connector on the backpanel and DB-9 male on a cable whose other end has the hermaphroditic connector.

The basic speed of the network is 4 million bits per second (Mbps), or 500,000 bytes per second. This does not include the many

levels of overhead in a typical application, and throughput for the user will often be many times less than that. There is nothing in the hardware or software architecture that limits the network to 4 Mbps, and implementations at 16 Mbps are expected in the near future.

## Logical Interconnection

Each interconnection cable contains two twisted pairs of wires. Although the stations and MAUs are cabled in a star-like fashion, the electrical effect of the special cables and connectors is to create a continuous ring from station to station. One twisted pair in the cable to each station is used to transmit to the *next* station in the ring, and the other pair is used to receive from the *previous* station. The ordering depends on how cables are plugged into the MAUs and how the MAUs are interconnected.

The operation of the ring depends on each station retransmitting data from its receive pair onto its transmit pair, regardless of whether that station is involved in the conversation. In order to insure that the ring is operational even if some stations are turned off, connecting a cable from a station to an MAU is not sufficient to cause that station to enter the ring; it also must send a DC voltage on its transmit pair which triggers a relay in the MAU. If power to the station fails, or if the cable to the station is disconnected at either end, the relay loses power and the ring bypasses that station.

When the relay is not powered, the cable to the station has its transmit pair connected to its own receive pair so that it may test the network adapter and cable. Prior to inserting itself onto the ring by supplying the relay voltage the network adapter sends several thousand data frames to itself to verify correct operation. This process, plus the network adapter self-test, may take 15 seconds or more.

## Access control

Only one station on the entire ring is allowed transmit data at a time. To control access, a 3-byte message giving permission to transmit, called the *free token*, continually circulates when there is no other traffic. Each idle station retransmits it as it is received. A station which wishes to transmit data waits for the token, and then sends its data instead of the token. When its data transmission is finished, it regenerates the token message. In addition to this simple *rotational priority* scheme, there are also ways to establish other priorities; all messages including the token contain both a 3-bit priority field for itself and a 3-bit reservation priority for a possible subsequent message.

A data message, called a *frame*, may be directed to a single destination station or any of various groups of stations. In all

cases the receiver of the message does *not* remove it from the ring; he simply makes a local copy of it and retransmits it, just as do stations that are not receiving the message. It is the originator of the message who is responsible for removing the message from the ring when it returns to him, and then replacing it with the token.

In visualizing the traffic flow on the network it is important to realize that most frames are much longer (in time) than the round-trip delay around the ring. Each station introduces a delay of less than 3 bit times when it is repeating data from its receive cable to its transmit cable, whether or not it is making a copy of the data. That delay for each station plus the cable propagation delays produce the total ring round-trip time; for a typical network of 50 stations it might be about 50 microseconds. A 1000 byte (8000 bit) frame takes 2000 microseconds to transmit, so the transmitter must be removing the beginning of the frame that has made the trip around the ring much before it has finished sending out the whole frame.

In normal operation the token is circulated and regenerated by the cooperative operation of all stations acting democratically. If the token is destroyed by transmission error or other fault (a station inserting or de-inserting from the ring typically destroys the token because of electrical noise created by the relay operation) it is the responsibility of a station designated as the *ring monitor* to notice the absence of the token and regenerate it. There is only one ring monitor on the network at a time, although every station is able to assume the role if needed. If the active monitor is disabled or leaves the ring a *monitor contention* process begins through which a new active monitor is elected by the remaining stations.

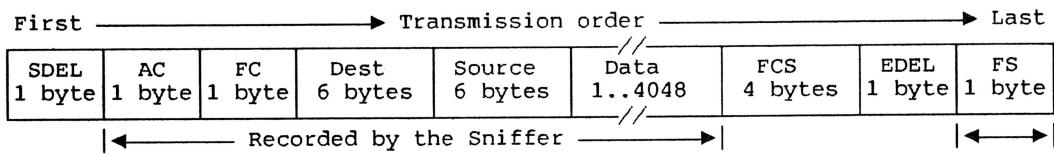
## Data format

All data is transmitted as a sequence of 8-bit bytes sent serially and Manchester encoded. The minimum transmission is the three byte token:

SDEL 1 byte	AC 1 byte	EDEL 1 byte
----------------	--------------	----------------

Both SDEL (starting delimiter) and EDEL (ending delimiter) have intentional Manchester code violations in certain bit positions so that the start and end of a frame can never be accidentally recognized in the middle of other data. The AC (access control) byte contains a bit which indicates that this is a token and not a data frame, and priority information. Tokens are not recorded by the Sniffer.

If a message is not a token then it is a data frame:



The SDEL, AC, and EDEL are as before, except the the AC now says that this is a data frame and not a token. The FC (frame control) byte contains some frame-type information. The destination and source addresses are each 6 bytes, and various kinds of broadcast addresses are indicated if the first bit of the destination address is a one. Following the data is a Frame Check Sequence (FCS) which checks the validity of all previous data starting with the AC bytes. The Sniffer records bytes starting with AC and ending with the last byte of the data field.

The last byte for data frames is the Frame Status (FS) byte which contains bits that may be set on by the recipient of the frame: *Address Recognized*, if a station matched the destination address, and *Frame Copied* if it was able to successfully make a local copy of the data as it passed by. Note that the FS is not covered by the Frame Check Sequence (so that the FCS doesn't have to be changed by the recipient), but for greater reliability the bits in the FS are each duplicated. The Sniffer records the FS byte and displays it in the detail window.

## Routing Information

There is an optional routing information (RI) field that may be present at the beginning of the data part of any frame. The RI field, which may be up to 32 bytes long, contains information about the path that the frame took if it was forwarded through multiple network segments by bridges. If the RI field is present, the first bit of the source address will be a one.

The RI field is not part of the IEEE 802.2 Token-Ring standard, although it has been proposed for official adoption. IBM bridges currently do use this extension to the standard.

## MAC frames

A data frame may be a MAC (Media Access Control) frame which contains information used to control the Token-Ring network itself. Most MAC frames are generated and processed by the computer's network adapter and are not of concern to software within the host. The type field in the FC byte indicates whether the frame is a MAC frame.

MAC frames are used for processes like monitor contention, error reporting and error recovery. In addition, the active monitor announces its presence with a periodic MAC frame, and all stations that could become the monitor should the need arise (*Standby monitors*) do likewise.

MAC frames contain a major type code followed by a variable number of variable-length fields called *subvectors* which give additional information.

## LLC frames

A data frame which is not a MAC frame is (in all non-proprietary uses of the Token-Ring network) a Logical Link Control frame. LLC is a protocol that provides reliable connection-oriented or connectionless virtual circuits between processes.

The data part of an LLC frame begins with a 3 or 4 byte header, the first two bytes of which are the Destination and Source Service Access Points (SAPs). The SAP numbers are preassigned codes which indicate which sub-protocol the rest of the frame belongs to. For example, the NETBIOS protocol has been allocated a single SAP (hex F0), and SNA has been allocated four SAPs (hex 04, 05, 08, and 0C). The Source and Destination SAPS (SSAP, DSAP) are almost always the same except in frames that are establishing an initial SNA connection.

There are three LLC frame types:

- *I* frames contain arbitrary sequenced data interpreted by the protocol that the SAPs designate. The LLC header contains a send sequence number N(S) for this frame and a receive sequence number N(R) of the next frame expected from the other station.
- *Supervisory* frames do not contain or consume an N(S) number, but do contain N(R). In addition they can contain the following indications as either a command or a response:

**RR**      Receive Ready; transmission can proceed.

**RNR**      Receive Not Ready; transmission is blocked.

**REJ**      Reject; retransmission starting with N(R) is requested.

- ***Unnumbered format*** frames contain neither N(S) nor N(R), but may contain control information or data. There are the following types:
 

<b>SABME</b>	Set Asynchronous Balanced Mode Extended; establish a virtual connection, also called a <i>link</i> .
<b>DISC</b>	Disconnect; terminate a virtual connection.
<b>DM</b>	Disconnected mode; the connection is broken.
<b>UA</b>	Unnumbered acknowledgement; for SABME and DISC response.
<b>FRMR</b>	Frame reject; the format of a received frame was bad.
<b>XID</b>	Exchange identification; for negotiation of LLC services.
<b>TEST</b>	Test probe; should be echoed by the receiving station.
<b>UI</b>	Unnumbered information; used by the SAP protocol for any purpose.

The only LLC frames which are allowed to contain data after the LLC header are I, UI, TEST, XID, and FRMR types.

Note that the minimal use of LLC is to make every frame a UI type, in which case the data part of every frame begins with the two SAP bytes and a UI (hex 03) control byte. This technique is typically used to implement various higher level protocols which do connection control and sequencing themselves and therefore do not need the services of the standard LLC.

## Appendix E. Glossary of Acronyms

<b>AC</b>	Access control; a DLC byte which contains the token indicator and frame priority information.
<b>ACTPU</b>	Activate Physical Unit; an SNA message sent to start a session.
<b>APPС</b>	Advanced Program-to-Program Communications; a communications system used to communicate between transaction programs on IBM computers; APPC uses the LU 6.2 subset of SNA.
<b>ASCII</b>	American Standard Code for Information Interchange; a mapping between numeric codes and graphical characters used almost universally for all personal computer and non-IBM mainframe applications.
<b>BIND</b>	An SNA message sent to activate a session between LUs.
<b>BIS</b>	Bracket Initiation Stopped; an SNA message sent to indicate that the sending station will not attempt to initiate any more brackets.
<b>CGA</b>	Color Graphics Adapter; The interface between a personal computer and a medium-resolution color monitor.
<b>DB-9</b>	A 9-pin standardized connector used in personal computers for the Token-Ring network connection, serial I/O port, and RGBI output.
<b>DB-25</b>	A 25-pin standardized connector used in personal computers for printer parallel output ports.
<b>DFC</b>	Data Flow Control; an SNA subprocess.
<b>DISC</b>	Disconnect; an LLC non-data frame indicating that the connection established by an earlier SABME is to be broken.
<b>DLC</b>	Data Link Control; the lowest protocol level of the Token Ring Network; fields include AC, FC, Destination address, and Source address.
<b>DM</b>	Disconnected mode; an LLC message acknowledging that a previously established connection has been broken.
<b>DOS</b>	Disk Operating System; the most common operating system for IBM-compatible personal computers.
<b>DSAP</b>	Destination Source Access Point; the SAP for the protocol expected to be used by the destination station in decoding the frame data.

<b>EBCDIC</b>	Extended Binary Coded Data Interchange Code; a mapping between numeric codes and graphical characters used for IBM mainframe computers and communications protocols defined by IBM.
<b>FC</b>	Frame control; a DLC byte which contains frame type information.
<b>FCS</b>	Frame check sequence; a 32-bit cyclical redundancy check field used to increase the probability of error-free transmission on the ring.
<b>FID</b>	Format Identification; a field in the SNA Transmission header indicating the type of nodes participating in the conversation. LU 6.2 nodes are type 2.
<b>FMD</b>	Function Management Data; a class of data embedded at the start of SNA RUs.
<b>FMH</b>	Function Management Header; the header part of SNA FMD containing addressing and transmission control information.
<b>FRMR</b>	Frame Reject; an LLC command or response indicating that a previous frame had a bad format and is being rejected. The REJ frame contains five bytes of data explaining why and how the previous frame was bad.
<b>FS</b>	Frame status; a byte appended to the frame following the checksum; contains the Address Recognized and Frame Copied bits.
<b>I</b>	Information; an LLC frame type used to send sequenced data which must be acknowledged.
<b>IEEE</b>	Institute of Electrical and Electronics Engineers, Inc. Standards documents are available from them at 345 East 47th Street, New York, NY 10017.
<b>LAN</b>	Local Area Network; the hardware and software used to connect computers together in limited geographical area.
<b>LLC</b>	Logical Link Control; the protocol level which initially defines the format of data frames sent on the Token Ring; standardized as IEEE 802.2 and ISO/DIS 8802/2.
<b>LSA</b>	Lost Subarea; an SNA error condition.
<b>LUSTAT</b>	Logical Unit Status; an SNA message used to send status information.
<b>LU 6.2</b>	Logical Unit 6.2; a subset of the SNA protocols used for peer-to-peer communications between computers.

<b>MAP</b>	Manufacturing Automation Protocol; an emerging multi-layer networking protocol developed primarily by General Motors for manufacturing control applications.
<b>MAU</b>	Multiple Access Unit; the wiring concentrator used for linking stations connected to the Token Ring.
<b>MAC</b>	Media Access Control; the protocol level which describes network management frames sent on the Token Ring. Most MAC frames are handled transparently by the network adapter.
<b>NC</b>	Network Control; an SNA subprocess.
<b>NETBIOS</b>	Network Basic I/O System; (1) A protocol implemented by the PC LAN Program to support symbolically named stations and arbitrary data. (2) The programming interface used to send and receive NETBIOS messages.
<b>N(R)</b>	Receive sequence number; an LLC field for I frames which indicates the sequence number of the next frame expected; all frames before N(R) are thus implicitly acknowledged.
<b>N(S)</b>	Send sequence number; an LLC field for I frames which indicates the sequence number of the current frame within the connection.
<b>PCF</b>	Physical Control Fields; the part of the DLC header including the AC and FC fields.
<b>PI</b>	Protocol Interpreter; a program which knows the frame format and transaction rules of a communications protocol and can decode and display frame data.
<b>REJ</b>	Reject; an LLC frame type which requests retransmission of previously sent frames.
<b>REM</b>	Ring Error Monitor; a station on the network which collects MAC-level error messages from the other stations.
<b>RGBI</b>	Red-Green-Blue-Intensity; a DB-9 connector convention used for attaching a color monitor to a personal computer.
<b>RH</b>	Request/response header; an SNA control field prior to a Request Unit or Response unit.
<b>RI</b>	Routing Indicator; the first bit in the source address field of a frame, which if 1 says that the data field begins with Routing Information.
<b>RNR</b>	Receive Not Ready; an LLC command or response indicating that transmission is blocked.

<b>RPS</b>	Ring Parameter Server; a station on the network which maintains MAC-level information about the LAN configuration such as ring numbers and physical location identifiers.
<b>RR</b>	Receive ready; an LLC non-data frame indicating readiness to receive data from the other station.
<b>RU</b>	Request Unit/Response unit; the part of an SNA frame after the RH which contains the details of a request or its response.
<b>SABME</b>	Set Asynchronous Balanced Mode Extended; an LLC non-data frame requesting the establishment of a connection over which numbered I frames may be sent.
<b>SAP</b>	Service Access Point; a small number used by convention or established by a standards group, which defines the format of subsequent LLC data; a means of demultiplexing alternative protocols supported by LLC.
<b>SBI</b>	Stop Bracket Initiation; an SNA message sent to request that the other station not initiate any more brackets.
<b>SC</b>	Session Control; an SNA subprocess.
<b>SDLC</b>	Synchronous Data Link Control; an older serial communications protocol which was the model for LLC and with which it shares many features.
<b>SIG</b>	Signal; a high-priority SNA message used to request permission to send.
<b>SMB</b>	Server Message Block; a message type used by the IBM PC LAN Program to make requests from a user station to a server and receive replies. Many of the functions are similar to those made by an application program to DOS running on a single computer. Under the IBM PC LAN Program SMBs are sent as data within NETBIOS frames.
<b>SNA</b>	Systems Network Architecture; a complex set of protocols used by IBM for network communications, particularly with mainframe computers.
<b>SPP</b>	Sequenced Packet Protocol; the subset of XNS which supports reliable connections using sequenced data.
<b>SSAP</b>	Source Service Access Point; the SAP for the protocol used by the originating station.
<b>SSCP</b>	System Services Control Point; an SNA identification of communications management functions.

<b>SUA</b>	Stored Upstream Address; the network address of a station's nearest upstream neighbor. Texas Instruments calls this the "UNA".
<b>TC</b>	Transmission control; an SNA subprocess.
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol; an older multi-layer networking protocol developed originally by the US Government for Arpanet and now used by several LAN manufacturers.
<b>TH</b>	Transmission header; the initial part of a SNA frame immediately following the LLC header.
<b>TS</b>	Transmission services; an SNA subprocess.
<b>UA</b>	Unnumbered Acknowledgment; an LLC frame which acknowledges a previous SABME or DISC request.
<b>UI</b>	Unnumbered Information; an LLC frame type used to send data without sequence numbers.
<b>UNA</b>	Upstream Neighbor Address; the network address of a station's nearest upstream neighbor. IBM calls this the "SUA".
<b>XID</b>	Exchange Identification; an LLC unnumbered frame type used to negotiate what LLC services will be used during a connection.
<b>XNS</b>	Xerox Network Systems; a family of protocols standardized by Xerox; in particular the Internet Transport Protocols. Documents are available from Xerox Office Products Division, 3333 Coyote Hill Road, Palo Alto, CA 94304.
<b>802.2</b>	The IEEE standards designation for the LLC Sublayer protocol.
<b>802.5</b>	The IEEE standards designation for the Token Ring access method.



## Appendix F. Sniffer Specifications

<b>Product Name:</b>	Token Ring Network Portable Protocol Analyzer.
<b>Model Number:</b>	PA-400.
<b>Package:</b>	Portable instrument with integral carrying case and handle.
<b>Weight:</b>	Approx 30 lbs in use, 45 lbs as shipped.
<b>Power:</b>	115 VAC 50/60 Hz, standard 6' US line cord. (230 VAC automatically switched option is available.)
<b>Processor:</b>	Intel 80286 at 8 Mhz.
<b>Memory:</b>	640 KB RAM.
<b>Storage:</b>	One 360 KB 5.25" internal floppy disk drive, One 20 MB internal hard disk.
<b>Display:</b>	9" internal green phosphor monitor, One DB-9 connector for external RGBI color monitor, One RCA connector for NTSC composite video monitor.
<b>Keyboard:</b>	Detachable typewriter-style keyboard with function keys.
<b>Network:</b>	One DB-9 token ring connector, plus 8' cable terminated in a standard IBM Token-Ring cable connector.
<b>Printer Port:</b>	One DB-25 connector for a parallel Centronics-compatible printer. One DB-9 connector for a serial printer.
<b>Protocols:</b>	DLC, RI, MAC, LLC, NETBIOS, SMB, and SNA protocol interpreters are standard. XNS, Novell Netware, Bridge, 3COM, Nestar, and others are available as options.
<b>Software:</b>	Written in a combination of microcode, machine language, and C. Source code is not provided, but end-user extensions for proprietary protocols are supported.
<b>Warranty:</b>	One year.
<b>Copyright:</b>	Software, manuals, and screen formats ("look and feel") are all copyright by Network General Corporation.



## **Appendix G. References**

Advanced Program-to-Program Communication for the IBM Personal Computer. Programming Guide. IBM Corporation, publication number 61X3842.

IBM PC Local Area Network Program. User's Guide, IBM Corporation, publication number 6139747.

IEEE Standards for Local Area Networks: Logical Link Control, ANSI/IEEE Std 802.2-1985, IEEE publication number SH09712.

IEEE Standards for Local Area Networks: Token Ring Access Method, ANSI/IEEE Std 802.5-1985, IEEE publication number SH09944.

PC Network Technical Reference, IBM Corporation, publication number 6322916.

Systems Network Architecture Reference Summary, IBM Corporation, publication number GA27-3136.

TMS380 Adapter Chipset User's Guide, Texas Instruments Incorporated, publication number SPWU001.

Token Ring Network Architecture Reference, IBM Corporation, publication number 6165877.

Token Ring Network PC Adapter Technical Reference, IBM Corporation, publication number 69X7713.



# Appendix H. Troubleshooting Checklist

Please fill in:

**Phone number, Network  
General's Customer  
Service Department:**

**Serial number  
of your Sniffer:**

## Check First

When you suspect a problem with the Sniffer equipment or software, please look through this check list before contacting us. Many times we find that correcting a simple oversight can save you (and us) lots of time.

**If there is nothing on the screen**

1. Check the power cable and power source.
2. Check the power switch (left side, back).
3. Make sure the monitor screen intensity control under the floppy disk drive is turned up (clockwise).

**If the Sniffer program does not start**

1. Make sure there is no diskette in the floppy drive when you start the Sniffer. (This may produce the message "Non-System disk or disk error.")
2. Doublecheck any modifications you might have made to the autoexec.bat startup file on the hard disk.

**If there is no output on the external color screen**

1. Make sure you answered "Y" to the question "Do you have an external color monitor installed?"
2. Make sure your monitor is attached to the slot 2 "Video" DB-9 connector and not the "Token Ring" DB-9 connector.
3. Check that the monitor is working and adjusted properly.

**If the keyboard or display is too low for comfort**

1. Extend the flaps under the keyboard, and erect the bottom support under the display unit.

***If you get the warning about  
“Not enough memory”, or  
“This is running on a slow  
computer”***

1. You may want to remove any resident programs you have installed, such as ProKey or Sidekick. They take both memory space and time.
2. Make sure you did not change the processing speed by using the CTRL-ALT-\ key combination.

***If you cannot capture data  
from the network***

1. The message “lobe media test failed” probably means that the Token Ring cable is not attached. Make sure it is plugged into the slot 3 “Token Ring” DB-9 connector and not the “Video” DB-9 connector.
2. The message “lobe wire fault” probably means the cable is not connected to a Multistation Access Unit (MAU).
3. If connecting to a 3COM Token Ring system, you must use a standard IBM Multistation Access Unit (MAU) port, and NOT the 3COM Ring-Tap unit.
4. Provided you’re using version 1.02 or later, make sure that the capture submenu says “From <Token Ring>.”

***If you don’t see traffic that you  
expect***

1. Check that you are not the only station inserted on the ring.
2. If there are multiple MAUs, check that they are cabled together properly.
3. Check the station, protocol, and pattern-match capture filters to see whether traffic is being discarded. If the “frames seen” count is larger than the “frames accepted” count, then frames are being discarded.

**If a pattern-match filter or trigger doesn't seem to work**

1. Doublecheck the offset; it is in DECIMAL, not HEX.
2. Check the offset origin: is it frame-relative (starting with the first frame byte) or data-relative (starting with the first LLC byte)?
3. Check the protocol and station-address filters; they may be causing the frames of interest to be discarded.

**If you get the message "No frames selected for display"**

1. Change the station address display filter, or
2. Change the protocol display filter, or
3. Change the pattern-match display filter.
4. Make sure that the display and capture filters are not mutually exclusive.

**If you don't see frames that you expect**

1. Check the display filters.
2. Check the capture filters, since they may have caused the frames to be discarded.

**If a "from" or "to" filter isn't working**

1. Check the setting of the "reverse direction" option in the filter menu.
2. Check the settings of other filters involved in capture or display; what you see displayed is what passes through *all* the filters.

**If the display seems inconsistent**

1. Press the HOME or END keys in the summary display, then return to the area you were looking at. If this corrects the problem, please tell us about it.

- If network utilization seems too low**
1. Check the display filters. Remember that only frames displayed are used in the network utilization calculation. Sometimes lower-level protocols which are not displayed (like NET) carry much of the actual data.
  2. Do some reasonableness checks based on the frame sizes. The utilization might indeed be correct!
- If you are not seeing symbolic station names**
1. Use the "Manage Names" menu item to examine and add to the names list.
  2. Remember to "Save Names" to make a permanent record of any changes.
  3. Make sure the "startup.trd" file is in the default directory, or in the directory specified by the SET TRNAMES= environment string of DOS.
- If HELP information is not available**
1. Make sure the "trsniff.hlp" file is in the default directory, or in the directory specified by the SET TRHELP= environment string of DOS.
  2. Make sure the various help files are in a subdirectory called "help" within the same directory as "trsniff.hlp."
- If you aren't getting any print output**
1. Check the printer itself (power, switch settings, etc.).
  2. Check the cable to the printer. For serial printers make sure you are connecting to the Sniffer's slot 1 male DB-9 connector and not one of the female DB-9 connectors.
  3. For serial printers, make sure that the transmit and receive pins are wired correctly. For some printers you may need a "null modem" cable.
  4. For serial printers, make sure that you have issued the appropriate MODE command to set the baud rate, word size, parity, etc.
  5. Check that you have selected LPT1 or COM1 (as appropriate) rather than "file" for the printer destination.
- If you aren't getting all the print output you expect**
1. Check the "from frame" and "to frame" options in the print menu.

Do you have "Frame nnn" selected instead of "First frame" or "Last frame"?
  2. Check the display filters; they affect what is printed. Remember that to print the entire detail report you must select all protocol levels AND turn off "highest level only."

***If you cannot save a trace to the disk***

1. Check the full pathname of the file being created. Is it the right drive letter? Do the intermediate subdirectories exist?
2. Make sure the hard disk or floppy disk has sufficient space for the file. The DOS command CHKDSK can be used for this purpose.

***If you have found a problem with the Sniffer software***

1. Try to recreate the problem after saving a trace file (perhaps filtered) and the setups, to a floppy diskette. Then:
  - Start the Sniffer.
  - Load the trace file.
  - Load the setups.
  - Recreate the problem.

If it is a display error, "print" the relevant part of the display to a diskette file.

If you can reconstruct the problem in this fashion, please send the diskette to the Network General Corporation Customer Service Department.

Be sure to include your system serial number (on the bottom of the right side connector panel) and the software version number (displayed in the initialization screen.)

***If your Sniffer hardware is malfunctioning***

1. See your warranty for more information, and contact the Network General Corporation Customer Service Department for information about obtaining repairs.



# Index

absolute time 97  
access control 122, 123  
access unit 121  
active monitor 24  
    present 14  
    Sniffer 24  
active window 92  
    zoom 94  
adapter  
    reset 55  
    token ring 1, 122  
address  
    filter 3, 56, 80  
    recognized 85 124  
    saved setup 105  
    station 15  
    symbolic names for 99  
    unrecognized 71  
addressee  
    NETBIOS command to seek 18  
    tabulation by 70  
alias for machine name 21  
angle brackets, modifier of station name 20  
ANSI standard 121  
Any station (in address filter) 57  
architecture  
    token ring 121  
arrow key 93  
    traversing menu 51  
ASCII  
    hexadecimal view 82  
AUTOEXEC.BAT 1, 46  
BACKUP  
    DOS command 49  
backup Sniffer software 45, 49  
bandwidth 30  
    network utilization estimate 71, 98  
bar graph 71  
batch file  
    example 26, 28  
    search for 30  
bits per second  
    network utilization 98  
bootable diskette 49  
brightness control 45  
broadcast  
    filter category 56  
    find message addressee 20  
    message 70  
    monitor present 15  
    station address 99  
buffer full  
    stopping criterion 67  
bytes  
    accepted count 68  
    per second 30, 121  
cable  
    Sniffer 42  
    token ring 1, 122  
capture  
    buffer full 67  
    continuous 67  
    elapsed time 68  
    filter 3, 55, 56  
    pause and resume 73  
    pause during 68  
    preparations 52  
    procedure 55  
    saved setup 105  
    start 67  
    stop 63  
    when to stop 65  
capture buffer  
    capacity 4  
    display 6, 75  
    format of saved data files 107  
    load with file of saved frames 75, 76  
    not in saved setup 105  
    position of trigger 65  
    print display 102  
    save to file 5, 6, 47, 74, 104  
CAPTURE directory 46  
Capture options function key 74  
CAPTURING (label on capture screen) 66  
CGA 43, 45  
change path 78  
character translation in hexadecimal view 82  
characters, resolution 45  
check marks in menu 53  
clear names 100  
clear screen function key 73  
close file  
    example 27  
color monitor 2, 43, 44  
    vs. monochrome 45  
COLOR parameter to TRSNIFF 46  
COM1 103

COMMAND.COM 46  
COMPAQ 41  
composite video 44  
connect  
    ring insertion 23  
    Sniffer to token ring 42  
contention 123  
context in summary view 6  
continuous capture 67  
count  
    bytes 4  
    data flow 26  
    frames 4  
    total bytes and frames 68  
    traffic by sender 69  
    traffic by station pairs 69  
create new directory 78  
current directory 47, 77  
Data display function key 73  
data flow metering 26, 67  
data format  
    token ring 123  
data format, saved data files 107  
data frame 124  
data-relative offset 62, 65  
date file created 35  
DB-9 connector 42  
    monitor vs. token ring 44  
DB-9 connector 2  
decimal vs. hex offset 62  
delay in delivering message 13  
delimiter  
    token ring 123  
delta time 97  
density, traffic 4, 28  
destination 124  
detail view 7, 18, 82, 84  
    highlight corresponding hex 93  
    protocol level 84, 85  
dictionary of station names 15, 17, 99  
DIR> in list of saved files 77  
directory  
    conventions 47  
    create new 78  
    current vs. path 48  
    path not in saved setup 105  
    path to different 48, 77  
    structure 46  
    to run Sniffer 47, 77  
disconnect from token ring 55  
disk drive 41  
diskette to run Sniffer 45, 49  
diskettes 44  
display  
    capture buffer 6  
    detail view 82, 84  
    form of 7  
    hexadecimal view 82  
    option in main menu 75  
    preparations 52  
    setting form 82  
    summary view 82, 87  
display filter 5  
    procedure to set 80  
    protocol 81  
    saved setup 105  
DLC  
    protocol level in detail view 85  
DOS 41  
    backup command 49  
    directory in which stored 46  
    environment 46  
    exit to 45, 49, 52  
downstream neighbor 23  
dual viewports 8  
duplicate address test 24  
EBCDIC  
    hexadecimal view 82  
edit names 17, 72, 101  
ending delimiter 123  
Enter key 53  
environment, DOS 46  
error monitor 25, 99  
error report  
    soft error 25  
example 13  
    batch file on server disk 26  
    insertion in ring 23  
    search for batch file 30  
    slow delivery of message 13  
    SMB frames 16  
exit to DOS 45, 52  
file  
    captured frames 74  
    example of remote use 27  
    handle 35  
    name conventions 111  
    print to 102  
    read 36  
    repeated open and close 28

saved frames 5, 104  
saved setup 105  
server 26  
size and date of creation 35  
filter  
    address 56  
    capture 3, 55, 56  
    display 5, 81  
    display, procedure to set 80  
    pattern 80  
    pattern match 3, 56  
    protocol 3, 56, 80  
    saved setup 105  
    station address 3, 80  
Find name  
    NETBIOS command 19  
floppy disk to run Sniffer 45, 49  
floppy drive 41  
format of saved capture files 107  
forwarded message 21  
frame 122  
    capture 55  
    context in summary view 6  
    copied 85  
    count total seen 68  
    display 6  
    filtered out of display 96  
    jump to 95  
    jump to mark 95  
    jump to number 96  
    jump to trigger 95  
    numbering 78  
    printing report of 102  
    scroll to next or previous 93  
    search for pattern 96  
    status 124  
    trigger 6  
    validity checks 85  
frame copied 124  
frame-relative offset 62, 65  
free token 23  
full buffer  
    stopping criterion 67  
function keys 53  
    capture phase 73  
go to frame 95  
graphic display of traffic rate 71  
half-byte character pattern 61  
handle 35  
hard disk 1, 42  
    organization of directories 46  
hardware 41  
heartbeat, monitor frames 14, 18  
HELP directory 46, 47  
help, on-line 54  
hexadecimal  
    highlight parts 93  
    message text visible 21  
    station address 71  
    view 7, 82  
    vs. decimal offset 62  
high-water (in transmission display) 67  
highest-level-only  
    difference between screen and printer views 103  
    protocol view 89  
    summary view 28  
highlight  
    active window 92  
    hex corresponding to detail interpretation 93  
home key 93  
IEEE standard 121  
initialization request 25  
insertion in ring 55, 122  
    example 23  
interpretation  
    frame 9  
    highlight corresponding hex 93  
    see also detail view 9  
interpreter  
    protocol 16, 115  
interval  
    between frames 97  
    between monitor broadcasts 15  
ISO standard 121  
jump  
    to frame 95  
    to menu item by typing first letter 76  
keyboard, Sniffer 42  
kilobytes accepted count 68  
LAN Manager 24  
LAN manager, station address 99  
level of protocol  
    detail view 84  
    print vs. screen 103  
    summary view 89  
LLC 17  
    frame 125  
    protocol level in detail view 85  
load data command 77  
logarithmic scale 71

logical address, NETBIOS protocol 20  
logical link control see LLC  
look for names 100  
LPT1 103  
MAC frames 14, 125  
  character translation 82  
  summary view 14  
main menu 44, 50, 52  
  figure 51  
make directory 78  
MAKEFLOP program 49  
manage names 99  
Manchester code 123  
mark  
  basis of relative time 98  
  jump to 95  
  set 19  
MAU 121  
media access control 125  
memory, Sniffer 42  
menu  
  check marks 53  
  jump to item by typing first letter 76  
  main 44  
  overview 1  
  radio controls 53  
Menus function key 74  
message  
  delay in delivering 13  
  example of transmission 21  
  NETBIOS command 18  
  phases in transmission 20  
  SMB command 16  
meter  
  data flow 26, 67  
  traffic density 4  
modifier, station name 20  
monitor  
  active 125  
  brightness control 45  
  built-in 2  
  color 2, 43  
  color vs. monochrome 45  
  contention 123  
  frames 14  
  monochrome 42  
  no picture 45  
  standby 125  
  station address 99  
move to frame 95  
multiple access unit 121  
names  
  deduced from network messages 100  
  dictionary 17  
  directories for 48  
  editing 72, 101  
  file 71  
  not in saved setup 105  
  revision 59  
  station addresses 15  
  symbolic 16, 99  
  updating entries 48  
  working copy vs. file 59  
  working table vs. STARTUP.TRD file 99  
NETBIOS  
  example 3  
  in LLC frame 28  
  in trigger pattern 63  
  logical address 20  
  No receive 37  
  No receive command 32  
  Receive outstanding 32  
network  
  Sniffer's role 2  
  topology 25, 121  
  utilization 29, 71, 98  
network utilization 71  
New Capture function key 67, 74  
new directory 78  
new station 24, 71  
New station (in address filter) 57  
next frame 93  
nibble character pattern 61  
no picture visible 45  
No receive NETBIOS command 32, 37  
numbering of frames 78, 96  
offset  
  frame-relative vs. data relative 65  
  hexadecimal vs. decimal notation 93  
  pattern 62  
  trigger menu 63  
  trigger pattern 64  
open file 27, 28  
options saved in setup file 52  
other SAP  
  capture filter 60  
  custom interpreter for 113  
  display filter 81  
packing list 41  
page up/down keys 93  
pairwise tabulation of traffic 69  
panels of main menu 51

param server 24  
path  
  different directory 48, 78  
  not in saved setup 105  
pattern  
  capture filter 61  
  filter 3, 56, 80  
  saved setup 105  
  search for frame 96  
  trigger 64  
  working copy vs. file 62  
pause  
  capture 68  
  function key 73  
  options during 73  
percentage utilization 71  
polling, station-to-station 18  
portable 42  
power switch 42  
pretrigger  
  percentage of frames in buffer 66  
previous frame 93  
print  
  capture buffer 5, 102  
  example 27  
  vs. screen display 103  
priority 122  
protocol  
  capture filter 60  
  custom interpreter 113  
  display filter 80  
  filter 3, 56  
  interpreter 16  
  interpreter registration 115  
  level in detail view 84  
  print vs. screen 103  
  saved setup 60  
purge ring 23  
radio controls in menus 53  
rate of data transmission 67, 121  
RCA connector 44  
read byte range  
  SMB command 36  
read file 36  
  example 27  
real-time display 4, 67  
receive outstanding, NETBIOS command 32  
relative time 15, 29, 97  
repeated transmission 37  
request initialization, MAC command 25  
reset adapter 55  
resolution, monitor 43, 45  
RESTORE, DOS command 49  
Resume function key 74  
reverse direction (station address filter) 56  
RGBI video 2, 44  
ring  
  disconnect 55  
  insertion 23, 55, 122  
  purge 23, 97  
ring architecture 121  
routing information 124  
SAP  
  capture filter 60  
  other 60, 81  
save  
  capture buffer 74, 104  
  format of files 107  
  names 99  
  setup 8, 105  
saved frames  
  load capture buffer 5, 75  
saved setup 60, 62, 105  
schematic view of Sniffer 9  
scrolling 7  
  active view 94  
  view of frame display 93  
search directory  
  SMB command 27, 31  
search for pattern 96  
  saved setup 105  
send message  
  SMB command 16  
sender  
  tabulation by 70  
SET TRNAMES 48  
setup  
  file 55, 67, 105  
  pattern saved with 62  
  save 52  
  save and restore 8  
  saved 60  
  values saved 105  
size of file reported 35  
slots, Sniffer 42  
SMB  
  embedded in NETBIOS 28  
  example 3, 16, 27, 28  
  read byte range command 36  
  search directory command 31  
  send message 16

SNA  
     character translation 82  
     protocol level in detail view 84  
 Sniffer  
     description 1  
     hardware 41  
     procedure to start 44  
     role on network 2  
     schematic view 9  
     software 44  
     symbolic name 100  
 soft error 25  
 software, Sniffer 44  
 source 124  
 standby monitor present 14  
 starting delimiter 123  
 STARTUP.TRD file 71, 99, 99  
 station address  
     filter 3, 56, 80  
     saved setup 105  
     symbolic equivalent 15, 16  
 station names  
     symbolic equivalent 99  
 status  
     frame 124  
 stopping capture  
     trigger menu 63  
 SUA (station upstream address) 25  
 summary view 7, 82, 87  
     example 14  
     protocol level 89  
     two station format 88  
 switch, on-off 42  
 T, trigger frame label 66  
 Tab key 92  
 test file, batch example 26  
 This Sniffer symbolic name 100  
 time  
     absolute 97  
     display of 97  
     interval between frames 97  
     network utilization 98  
     relative 15, 29, 97  
 token 122  
     lost 23  
 token ring  
     architecture 121  
     connection 55  
     connector 42  
     disconnect from 55  
     insertion 55  
     speed 121  
     standards 121  
 topology 121  
     deduced from upstream neighbor 25  
 traffic  
     by station pairs 26  
     density 71  
     measurement 27, 67  
     tabulation by sender 70  
     tabulation by sender and addressee 4, 70  
     tabulation by senders 4  
     units to count 71  
 transmission rate 67  
 TRC, file extension 76, 111  
 trigger 55  
     detector to scan arriving frames 4  
     effect 5  
     frame 6  
     half-byte character pattern 64  
     offset to pattern 64  
     pattern match 63  
     procedure to set 6  
     saved setup 105  
 trigger frame  
     jump to 95  
     position in capture buffer 65  
 TRIGGERED (label on capture screen) 66  
 TRS, file extension 111  
 TRSNIFF.EXE 46, 77  
 TRSNIFF.HLP 111  
 two viewports 8, 22, 90  
 two-station format 8, 17, 27, 88  
     selection of stations 88  
 upstream neighbor 23, 24  
 utilization of network 29, 71, 98  
 validity checks 85  
 version number, in saved data files 108  
 view  
     alternative in display of data 5  
     decimal 7  
     detail 82, 84  
     difference between screen and printer 103  
     different frames 90  
     hexadecimal 7, 82  
     multiple windows 86  
     scrolling 93, 94  
     summary 7, 82, 87  
     zoom in active window 94  
 viewports, two 8, 90

**window**

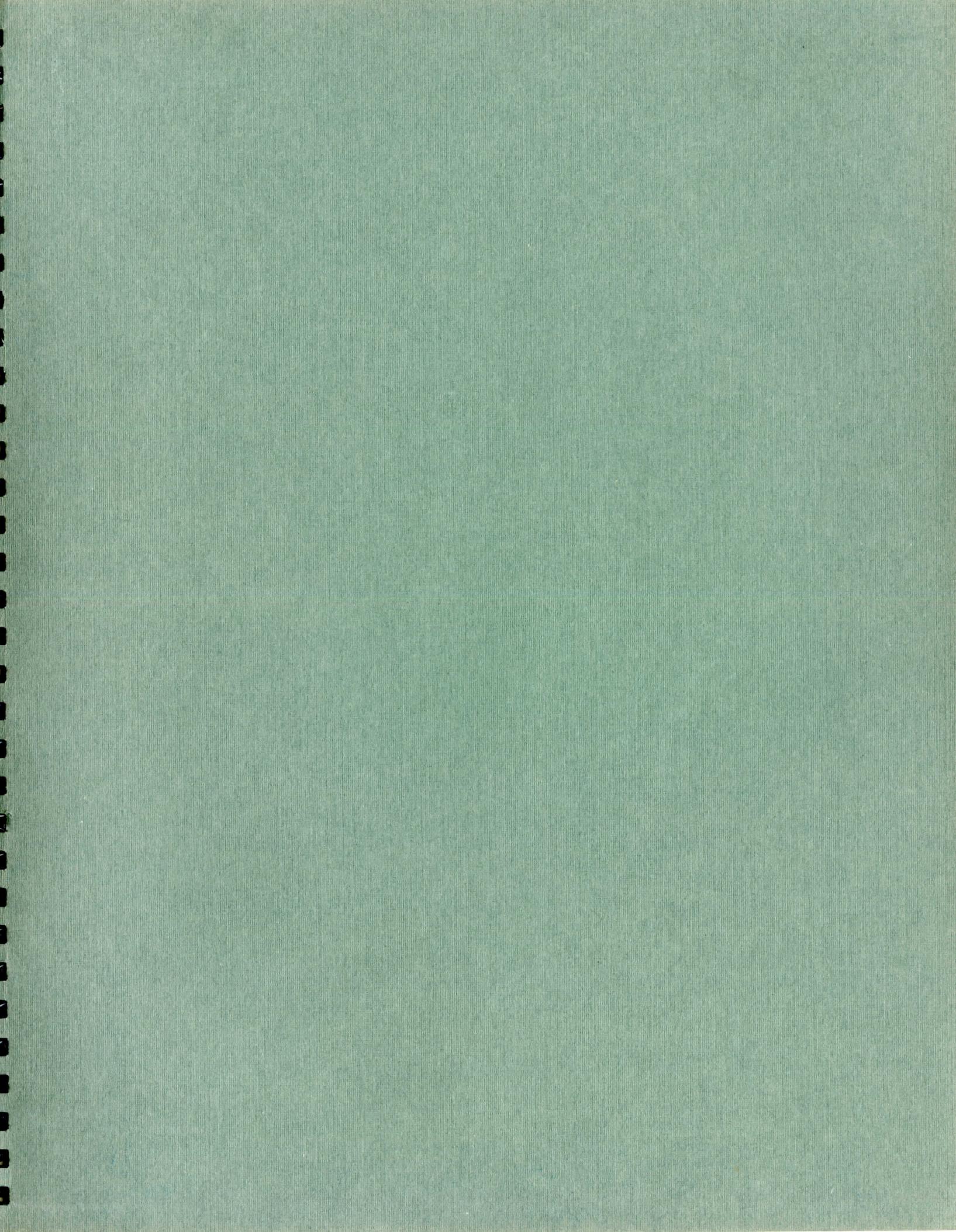
- active 92
- display 7
- highlight 93
- multiple views 86

**XX**

- code for “don’t check” 65

**zoom view of active window** 7, 35, 94





**Network General Corporation**  
1296B Lawrence Station Road  
Sunnyvale, California 94089

Part No. 20002-002  
Printed in U.S.A.